

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено

Завідувач кафедри

О.В.Коваль

(підпис)

(ініціали, прізвище)

“ ” 2019 р.

**ДИПЛОМНА РОБОТА**  
**на здобуття ступеня бакалавра**

з напрямку підготовки  
6.050101 “Комп’ютерні науки”

на тему: Прогнозування на основі теорії фракталів

Виконав: студент 4 курсу, групи ТМ-52

Салун Олександр Денисович

(прізвище, ім’я, по батькові)

(підпис)

Керівник д.т.н., професор Аушева Н.М.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент доц.,к.т.н., доцент Шевель Є.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 6.050101 “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ О.В. Коваль  
(підпис)

” \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

Салуну Олександру Денисовичу

(прізвище, ім’я, по батькові)

1. Тема роботи \_\_\_\_\_ “Прогнозування на основі теорії фракталів”

керівник роботи \_\_\_\_\_ д.т.н., професор Аушева Наталія Миколаївна

(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ” \_\_\_\_ ” \_\_\_\_\_ 201\_\_ р.

№ \_\_\_\_\_

2. Строк подання студентом роботи \_\_\_\_\_ 201\_\_ р.

3. Вихідні дані до роботи сторінка з проаналізованим графіком.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати сучасні методи для аналізу та прогнозування процесів, проаналізувати програмне забезпечення для розв’язання задач прогнозування, розробити програмне забезпечення, розробити концептуальну модель бд, розробити інтерфейс користувача.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

5. Перелік ілюстраційного матеріалу

1. Мета та завдання роботи 2.Огляд існуючих рішень 3. Програмні засоби для прогнозування процесів4.Фрактальний аналіз ринків 5. Використані програмні засоби  
6.Висновки

Дата видачі завдання ” 14 ” жовтня 2019 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі	09.10.2019	
2.	Розробка архітектури та загальної структури системи	14.10.2019-23.12.2019	
3.	Розробка структур окремих підсистем	02.02.2019-03.03.2019	
4.	Підготовка матеріалів	04.03.2019-14.04.2019	
5.	Програмна реалізація системи	15.04.2019-19.05.2019	
6.	Оформлення пояснювальної записки	20.05.2019-05.06.2019	
7.	Захист програмного продукту	14.05.2019	
8.	Передзахист	28.05.2019	
9.	Захист	17.06.2019-21.06.2019	

Студент

\_\_\_\_\_  
(підпис)

Салун О.Д.

\_\_\_\_\_  
(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_  
(підпис)

Аушева Н.М.

\_\_\_\_\_  
(прізвище та ініціали)

## **АНОТАЦІЯ**

Мета дипломної роботи полягає в створенні веб-сервісу для високоефективного прогнозування економічних процесів на основі теорії фракталів. Користувацький інтерфейс дозволяє використовувати систему з електронних пристроїв будь-якого типу, що мають можливість використовувати web-браузер. Користувачі можуть створювати компанії з початковими даними, що використовуються для збору статистики та кінцевому прогнозуванню фінансового ринку. Система забезпечує клієнта отриманням результату та володінням повної інформації про рентабельність компанії в майбутньому. Записка містить 64 сторінки, 23 рисунків та 8 посилань.

## **ABSTRACT**

The purpose of the work is to create a web service for high-effective forecasting of economic processes based on the theory of fractals. The permission user interface uses a system of electronic devices of any type that can use a web browser. Users can create companies with initial data that are used to collect statistics and finite financial market forecasting. Customer accounting system for obtaining the result and possession of complete information about the profitability of the company in the future. The note contains 64 pages, 23 figures and 8 links.

# ЗМІСТ

## Вступ

1. Постановка задачі прогнозування економічних процесів
2. Існуючі методи та програмні засоби для прогнозування процесів
  - 2.1. Методи прогнозування
  - 2.2. Аналіз наявних інструментів
3. Реалізація задачі прогнозування на основі теорії фракталів
  - 3.1. Поняття теорії фракталів для задачі прогнозування
  - 3.2. Розробка методики використання теорії фракталів в задачі прогнозування
  - 3.3. Побудова алгоритму прогнозування на основі теорії фракталів
4. Засоби реалізації програмного забезпечення
  - 4.1. Вибір середовища створення програмного продукту
  - 4.2. Архітектура програмного забезпечення
  - 4.3. Діаграма класів
  - 4.4 Структура БД
5. Інтерфейс роботи з користувачем
  - 5.1 Системні вимоги
  - 5.2 Робота користувача з програмною системою

## Висновки

## Список використаних джерел

## Додаток А

## Додаток Б

## Додаток В

# **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

ПЗ – програмне забезпечення

JS – JavaScript

## **ВСТУП**

Актуальність теми. Прогнозування в наш час відіграє дуже важливу роль, так як соціально-економічні процеси постійно ростуть і необхідний точний аналіз. Крім того, тимчасові ряди, що описують динаміку різних складних систем штучного і природного походження, часто випускалися великомасштабними і самоподібними. Це може виражатися в збереженні структури ряду з його аппроксимацией, повторенням динаміки системи з різними часовими інтервалами тривалості, складної частотою і позірним нестационарним числом. Ці характеристики є ключовими для систем спеціального типу, в літературі званих фрактальними, великомасштабними та самоподібними.

Дослідження фрактальних часових рядів були розпочаті Мандельбротом в 1977 році і продовжені в працях Е. Фідер, М. Шредера, М. Кроновера, М. Ячменю. В даний час спостерігається тенденція до галузевого поділу дослідження великомасштабних часових рядів. Фрактали в економіці вивчають Е.Пітерс,

М.Дубовіков, М.Такаясу, В.Чоу, Д.Сорнетт. Великий внесок у розвиток фрактальних сигналів в радіоелектроніці внесли А.Потапов, В. Кравченко, В.Болотов, О.Лазоренко, Г.Ворнел.

Особливості фрактальної системи дозволяють прогнозувати і моделювати генеруючу систему, оцінювати її характеристики (наявні і бажані), генерувати самоподібні дані для потреб користувача (побудова рекаустіческіх кривих, синтез складних сигналів).

Такі завдання часто пов'язані з прийняттям рішень або розпізнаванням образів, тому перспектив включення в інтелектуальні системи підтримки прийняття рішень (наприклад, системи управління інвестиційними фондами чи моніторингу пацієнтів) підсистеми широкомасштабного аналізу даних багатообіцяюче. Серед прихильників такого підходу варто відзначити О. Кастільо, П. Меліна, А. К. палить, М. Саркара.

На сьогоднішній день існує велика кількість методів і моделей для аналізу багатомасштабних табличних часових рядів: як загальних, так і секторальних. Зокрема, фрактальний броунівський рух, вейвлетного моделі, системи ітерових функцій. Але всі вони мають свої недоліки: перший - аналітична складність і вузькість підкласу, описаного в великомасштабному тимчасовому ряду; другий - непридатність і грубість; третій - гарне рішення прямої задачі (синтез), але зовсім не вирішує зворотній (аналіз).

Тому ставиться завдання створення універсальної і простої моделі, яка легко обчислюється на комп'ютері і модифікується відповідно до потреб конкретного дослідження, зручною у практичній роботі (при оцінці параметрів і моделюванні фрактальних процесів довільних форм).

# 1. ПОСТАНОВКА ЗАДАЧІ ПРОГНОЗУВАННЯ ЕКОНОМІЧНИХ ПРОЦЕСІВ

Метою дипломної роботи є створення програмного забезпечення для аналізу та прогнозування соціально-економічних процесів.

Об'єктом дослідження є процеси аналізу і синтезу образів часових рядів та фінансових ринків.

Предметом дослідження є соціально-економічні показники(графіки та тренди) та моделі багатомасштабних часових рядів, методи розпізнання, відтворення та прогнозування їх образів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати сучасні методи для аналізу та прогнозування процесів, багатомасштабних часових рядів та застосування сплайнів і фракталів
- проаналізувати програмне забезпечення для розв'язання задач прогнозування;
- обрати засоби реалізації для прогнозування соціально-економічних процесів і алгоритм побудови різних видів фрактальних сплайнів;
- розробити структуру або архітектуру програмного забезпечення;
- розробити концептуальну модель бази даних;
- розробити програмне забезпечення і провести моделювання розроблених алгоритмів аналізу.

Поставлені у роботі задачі будуть розв'язані на основі фрактального методу.

Необхідно розробити програмне забезпечення для аналізу соціально-економічних процесів на основі теорії фракталів.

Вхідною інформацією вважаються наступні дані:

- Статичні данні з графіків різних компаній.
- коефіцієнти Херста для аналізу графіка;



Вихідною інформацію вважаються наступні дані:

- змодельований графік;
- змодельований тренд;
- змодельований прогноз ціни в подальший час;

При проведенні досліджень у дипломній роботі використовувалися методи: фрактальної геометрії, теорії сплайн-функцій, цифрової обробки сигналів та теорії алгоритмів. Для експериментальних досліджень використано тренд будь-якої компанії, метод обробки інформації та розроблено програмне забезпечення в середовищі Java Script.

## **2. Існуючі методи та програмні засоби для прогнозування процесів**

В даному розділі йдеться про існуючі методи прогнозування та програмні засоби. На даний момент існує багато методів прогнозування, кожен з них має багато формул для прогнозування, що стає складним в користуванні і не кожен метод здатний прогнозувати все дуже точно.

### **2.1 Методи прогнозування**

Залежно від форм роботи з експертами, методи експертного прогнозування можна розділити на дві групи: методи індивідуальної експертизи і методи колективної експертної оцінки.

Методи індивідуальної експертизи оцінки - методи прогнозування, основані на використанні в якості джерела інформації одного експерта. Найчастіше використовують два методи формування прогнозу: інтерв'ю і аналітичне експертна оцінка.

Метод інтерв'ю передбачає діалог прогнозіста з експертом, в ході якої він відповідно до заздалегідь розробленого продукту ставить перед ним питання щодо перспектив розвитку об'єкта. Схема діалогу: питання-відповідь. При цьому експерт використовує в основному тільки апріорними уявленнями про об'єкт прогнозування. Успіх такої оцінки в значній мірі залежить від здатності давати певні відповіді на питання один одному.

Метод експертних аналітичних оцінок оснований на отриманні оцінок прогнозованого об'єкта шляхом логічного аналізу. Аналітичні експертні оцінки передбачають тривалу і виконавчу роботу експерта над аналізом тренду, шляхів розвитку об'єкта прогнозування та оцінкою стану. Цей метод надає можливість експерту використовувати всю необхідну інформацію про об'єкт прогнозування. Висновки експерт оформляє у формі записки чи листа.

Основним плюсом цих методів є можливість максимального використання

індивідуальних здібностей фахівця, а також незначне психологічний тиск на фахівця. Але ці методи не підходять для прогнозування загальних стратегій через обмежені знання одного фахівця.

З метою підвищення обґрунтованості прогнозів для їх розробки залучаються декілька експертів, оцінки яких зіставляються і збираються між собою, створюючи колективну оцінку.

Методи колективної експертної оцінки ґрунтуються на виявленні узагальненої оцінки експертної групи шляхом обробки індивідуальних незалежних оцінок експертів в групі та аналізу.

В основі застосування методів колективної експертизи лежить гіпотеза про наявність у експертів здатності з достатнім ступенем вірогідності оцінювати: знання і важливість проблеми параметра, фактора, напрямки розвитку, характеристик і т.д., часу тієї чи іншої події; значення прогнозованих параметрів; доцільність вибору одного з альтернативних шляхів розвитку об'єкта прогнозування і т.д.

Методи колективної експертної оцінки на основі методу отримання інформації від експертів можна розділити на 2 великі групи: методи групової експертизи та методи анкетування.

Найчастіше використовуються такі методи групової експертизи: експертні комісії, колективна генерація ідей (метод "мозкової атаки"), оперативні гри.

Методи роботи експертних комісій колективної експертизи полягає в безперервній роботі експертів, згрупованих в групи, які розробляють лист про перспективи розвитку об'єкта прогнозування.

Метод генерації ідей ( "мозкові атаки") заснований на стимулюванні творчої діяльності експертів шляхом спільного обговорення конкретної проблеми. У той же час, обговорення регулюється певними правилами: забороняється оцінювати висунуті ідеї; час обмежений одним доповідачем; допускаються виступи кожного експерта кілька разів; пріоритет має експерт, який розробляє попередню ідею; обов'язково фіксуються всі ідеї; оцінка висунутих ідей здійснюється на наступних етапах.

Існує три різновиди методу колективної генерації ідей ( "атаки мозку"): метод

прямої групової експертизи ( "прямий мозкової атаки"), метод "групового консенсусу" і метод "оперативного творчості".

Метод прямої групової експертизи заснований на гіпотезі, що серед ідей, висунутих експертами, є принаймні, кілька хороших. Процедура проведення прямої групової експертизи виглядає наступним чином: група експертів з різних спеціальностей є проблемою, яка може мати кілька варіантів рішення. Мета цього методу - зібрати ідеї з поставленої нами проблеми. Потім, в кілька етапів, вони аналізуються, виключаються, що не вибираються і вибираються найбільш раціонально. Найчастіше цей метод використовується на рівні стратегічних і зовнішньополітичних рішень, а також у військово-економічному та військово-технічному прогнозуванні.

Метод "групового згоди" заснований на проведенні групового дослідження з метою визначення згоди і компромісу думок по прогнозованому питанню, в якому беруть участь понад шість експертів. Встановлено, що збільшення числа експертів більш ніж на шість осіб призводить до збільшення часу і ускладнює негативну узгодженість багатьох думок, що не призводить до значного збільшення кількості пропонованих ідей, якщо кількість експертів може бути враховано в повній мірі при обставинах, що склалися і знижує ступінь точності прогнозу. Метод "оперативного творчості" припускає, що сутнісні проблеми в цілому і можливі підходи до їх вирішення відомі тільки керівнику робіт. У цьому випадку необхідний експертний опитування, щоб переконатися, що підхід до проблеми правильний і що можна бути впевненим у своїй роботі.

У практиці прогнозування у воєнній сфері широко використовується методика оперативних ігор в формі навчань. Експерти не тільки дають оцінку ситуації, але і приймають рішення, виступаючи в якості командирів прийняття рішень в реальних бойових умовах, а також офіцерів штабу. Важливо, щоб в грі в ігри перевірялася правильність і правильність рішень, які приймаються в певних умовах. Досвід навчання за участю кваліфікованих фахівців дозволяє припустити, що експертне прогнозування з використанням таких повномасштабних моделей сприяє систематичному охопленню істотних елементів об'єктів і процесів прогнозування і

дозволяє отримати узагальнені рекомендації поточного і прогнозованого характеру.

У військовій грі, як правило, беруть участь обидві сторони конфлікту. Рішення приймаються відповідно до певних правил, які регулюються статутами і інструкціями. Формальна структура ігор побудована таким чином, що основні рішення, висновки, пропозиції підлягають критичному аналізу і аналізу. Використання озброєння і військової техніки, в тому числі автоматизованих систем управління, для ведення стрільби дозволяє нам приймати обґрунтовані рішення з питань, передбачених в плані досліджень.

Методи анкетування - методи колективної експертної оцінки, в яких анкети використовуються для анкетування експертів. Анкети можуть включати: питання, коли експертам необхідно дати однозначну відповідь на стан об'єкта, що проектується; заяву про допустиму майбутній картині деяких подій, а експерту потрібно тільки підтвердити або спростувати їх; надіслати листа з вказівкою оцінку факторів (ознак, параметрів, напрямків розвитку та т.д.), кількісну значимість прогнозованого показника або кордону, в якій він може перебувати в певний момент в майбутньому.

Експертні інтерв'ю проводяться з використанням анкет. В анкеті пропонується дати кількісну оцінку кожному фактору (ознакою, параметру, напрямку розвитку і т.д.), що є частиною завдання прогнозування. Наприклад, ми позначаємо ці фактори  $Y_1, Y_2, \dots, Y_j, \dots$  Зелень. Кожен з цих факторів,  $Y_j$ , може мати тільки якісне значення або мати кількісне вираження (наприклад, величину прогнозованого параметра). В обох випадках експерт повинен дати кількісну оцінку. Для факторів якісної достовірності така оцінка носить характер кількісного порівняння важливості цих факторів, а саме рангу або оцінки за певною шкалою. Для факторів (параметрів) кількісної достовірності оцінка дається числом, відповідним пропонованого значенням цього фактора (параметра).

З методів постановки питань при вирішенні військово-специфічних завдань використовуються найбільш поширені методи: метод оцінки за шкалою, метод парних звірень і метод MeDod Delphi ...

Обробка результатів експертного опитування залежить від типу інформації, отриманої від експертів.

Якщо кожен з  $m$  експертів, що беруть участь в опитуванні, відповідь на одне питання анкети, значення  $C_{ij}$  ( $i$  - число експертів) прогнозованого значення  $j$ , то при обробці  $m$  значень  $C_{ij}$  можуть бути обчислені наступні ключові моменти:

середнє значення експертних оцінок (точковий прогноз), що характеризує загальну думку експертів:

$$\tilde{M}[Y_j] = \frac{1}{m} \sum_{i=1}^m C_{ij} ;$$

розбіжність оцінок, що характеризує розкид думок (спот-прогноз) експертів щодо середнього значення::

$$\tilde{D}[Y_j] = \frac{1}{m-1} \sum_{i=1}^m (C_{ij} - \tilde{M}[Y_j])^2 ;$$

середнє квадратичне відхилення:

$$\tilde{\sigma}[Y_j] = \sqrt{\tilde{D}[Y_j]} ;$$

коефіцієнт варіації, що характеризує ступінь однобічності експертів при оцінці коефіцієнта  $j$  (параметр):

$$V_j = \frac{\tilde{\sigma}[Y_j]}{\tilde{M}[Y_j]} .$$

Чим вище коефіцієнт  $V$ , тим більше збігається думка експертів.

Параметри  $M_j$  і  $j$  дозволяють дослідити інтервальний прогноз. Для цього визначається розмір ділянки, на який падає ймовірна вартість майбутньої вартості прогнозованого значення:

$$\tilde{M}[Y_j] - \varepsilon_1 \leq C_j \leq \tilde{M}[Y_j] + \varepsilon_2 .$$

Значення, що визначають довірчий інтервал 1 і 2, залежать від величини довірчої ймовірності і закону розподілу суми кількостей і розраховуються відповідно до правил, викладених у третьому розділі підручника. Отже, якщо закон розподілу можна вважати нормальним, то для  $j$ -го прогнозованого значення:

$$\varepsilon_1 = \varepsilon_2 = t_\beta \frac{\tilde{\sigma}[Y_j]}{\sqrt{m}} ;$$

де  $t_\beta = \arg \Phi^* \left( \frac{1+\beta}{2} \right)$  – величина, яка обернена нормальній функції розподілу

$\Phi^*(x)$ , вираховується для даного значення імовірності  $\beta$ .

При обробці експертних даних також проводиться оцінка суперечливого думки експертів. Розглянемо поняття "протиріччя" думки колективного експерименту узагальненому думку всіх експертів. Припустимо, що думка  $k$ -го експерта є екстремальним серед думок  $m$  експертів. У зв'язку з тим, що фактичне значення дисперсії зазвичай невідомо, але відома тільки її оцінка, то для математичної оцінки суперечливого думки  $k$ -го експерта ми розраховуємо ймовірність того, що значення дисперсії буде більше за певну кількість  $\gamma$ :

$$\alpha = P \left( \frac{C_k - \tilde{M}[Y_j]}{\tilde{\sigma}[Y_j]} > \gamma \right).$$

Якщо ця ймовірність досить велика (наприклад, більше 0.05 - 0.10), то гіпотеза про аномалії може бути відкинута, в іншому випадку вона приймається. В цьому відношенні це протиріччя, в якому здійснюється нерівність  $C_{kj}$ , при якій виконується нерівність

$$C_{kj} - \tilde{M}_j[Y_j] > \gamma \tilde{\sigma}[Y_j] \quad (1.1)$$

з імовірністю, меншою значення  $\alpha'$ . Значення береться таким, що  $\alpha' = 0,1 - 0,05$ . Значення коефіцієнта, що є в умові (1.1), наведені в таблиці. 1.1.

Таблиця 1.1. Імовірність значень коефіцієнта  $\gamma$

$m$	$\alpha$		
	0,10	0,05	0,01
7	1,7	1,8	1,9
9	1,9	2,0	2,2
11	2,0	2,1	2,4

13	2,1	2,2	2,5
15	2,2	2,3	2,6
17	2,2	2,4	2,7
19	2,3	2,3	2,8
21	2,3	2,5	2,8

Виконання умови (1.1) з  $\leq$  математичним знаком існування підокруги мислення серед групи експертів. Слід підкреслити, що цей знак можна використовувати тільки в тому випадку, якщо розподіл оцінок експертів можна вважати нормальним.

Якщо екстремальне значення (точковий прогноз колективного експерта) буде підмножиною, то наступне найближче значення буде перевірятися до тих пір, поки таке значення не стане узгодженим.

Виконання умови протиріччя (або суперечності) залежить від величини ймовірності, тому дана ознака є умовним і повинен бути доповнений логічним аналізом, що враховує вимоги до точності прогнозу. При обробці результатів експертних опитувань слід також мати на увазі, що колектив найкращим чином представляє розвиток проєктованого об'єкта (процесу) в травні-майбутньому і тому "випадає" з області, що характеризує думки його колег. Тому до екстремальних значень експертних оцінок слід підходити дуже обережно.

Таким чином, послідовність оцінки результатів експертизи щодо майбутньої вартості виглядає наступним чином:

визначається загальна думка експертів (точкові прогнози);

визначені дисперсія і середнє квадратичне відхилення експертних оцінок;

оцінка протиріччя екстремальних оцінок проводиться за допомогою логічного аналізу і умов (1.1);

в ході консенсусу думок експертів результати опитування публікуються у вигляді точкових і інтервальних прогнозів;

У суперечливих думках проводиться другий раунд опитувань (з обговоренням результатів першого раунду).



Часто ця форма оцінки використовується, коли кожен  $m$  експертів дає по два (мінімальне  $C$  і максимальне  $C$ ) значення, між якими, на думку експерта, буде майбутнім значенням прогнозованого значення. Для обробки результатів опитування в першу чергу необхідно прийняти форму закону розподілу прогнозованого значення між екстремальними оцінками кожного експерта. Наприклад, закон рівномірного розподілу може бути обраний як такого закону пропорційного матеріального розподілу:

$$f(C_{ij}) = \frac{1}{C_{ij}^{\max} - C_{ij}^{\min}}, C_j^{\max} \leq C_j \leq C_j^{\min}.$$

В цьому випадку середнє значення (точковий прогноз), отримане від  $y$ - $m$ -експерта, визначається за формулою

$$\bar{C}_{ij} = \frac{1}{2} * (C_{ij}^{\max} + C_{ij}^{\min}).$$

Бальний прогноз за результатами узагальнення думок всіх експертів визначається за формулою

$$\tilde{M}_j = \frac{1}{m} \sum_{i=1}^m \bar{C}_{ij}.$$

Метод оцінки за шкалою. Може бути використаний як для факторів, що мають тільки якісну визначеність, так і для параметричних факторів, що визначаються цифрами.

Для факторів якісного детермінізму метод оцінки за шкалою передбачає нарахування балів за певною шкалою (100-бальною).

Вводимо наступну позначку:

$m$  - кількість експертів, які взяли участь у колективній експертизі; 1, 2, ...,  $i$ , ...,  $m$  - кількість експертів;

$n$  - число факторів; 1, 2, ...,  $j$ , ...,  $n$  - число факторів;

$m_j$  - кількість експертів, які оцінили  $j$ -фактор;

$c_{ij}$  - оцінка (в балах) відносної важливості  $j$  фактора і  $m$  експерта;

$m100j$  - кількість можливих міток (100 балів), отриманих  $j$ -м фактором.

Результати опитування заносяться в таблицю (таблиця. 1.2).

Таблиця 1.2. Результатів шкальних оцінок

Фактори (параметри)	Експерти					
	1	2	...	$i$	...	$m$
1	$C_{11}$	$C_{21}$	...	$C_{i1}$	...	$C_{m1}$
2	$C_{12}$	$C_{22}$	...	$C_{i2}$	...	$C_{m2}$
...	...	...	...	...	...	...
$J$	$C_{1j}$	$C_{2j}$	...	$C_{ij}$	...	$C_{mj}$
...	...	...	...	...	...	...
$N$	$C_{1n}$	$C_{2n}$	...	$C_{in}$	...	$C_{mn}$

Для оцінки важливості факторів вводяться характеризують їх показники:

загальна думка експертної групи про питому вагу факторів;

ступінь узгодженості думок експертів;

"Активність" експертів (ступінь їх участі в оцінці факторів);

компетентність експертів по кожному з факторів.

Індикатором узагальненого думки є середнє статистичне значення  $M_j$  оцінки кожного з чинників (в пунктах):

$$M_j = \frac{1}{m_j} \sum_{i=1}^{m_j} C_{ij}.$$

Це базовий показник, який може варіюватися від 0 до 100 і визначається для кожного з  $n$  факторів. Чим більше значення  $M_j$ , тим більше значення має  $j$ -фактор.

Інший підхід використовується для обробки експертних даних з метою визначення відносної важливості факторів. По-перше, обчислюється значення

$$W_{ij} = \frac{C_{ij}}{\sum_{j=1}^m C_{ij}},$$

де  $C_{ij}$  – оцінка  $j$ -му фактору  $i$ -м експертом;

$W_{ij}$  – маса  $j$ -го фактора за даними експерта.

Маса факторів всіх експертів визначається за формулою

$$W_j = \frac{\sum_{i=1}^m W_{ij}}{\sum_{i=1}^m \sum_{j=1}^n W_{ij}}.$$

Частота  $k_{100j}$  - це максимально можлива оцінка (100 балів), отримана  $j$ -м фактором

$$k_{100j} = \frac{m_{100j}}{m_j}$$

Даний показник вважається доповнюючим до основного і характеризує перевага фактора  $j$  з точки зору кількості "перших місць", відведених йому.

Поряд із середніми статистичними показниками ( $M$ ) і ( $k_{100j}$ ) максимально можливих оцінок для оцінки значущості даного чинника використовується сума оцінок від рангу до рангу, отриманих  $j$  фактором ( $S$ ), яка також використовується для визначення ступеня узгодженості думок експертів.

Для визначення суми рейтингів ранжування факторів проводиться шляхом зменшення оцінок, наданих кожному фактору кожним експертом: кожен бал, який присвоюється фактором  $m$ -експерт, є кількість цілих чисел таким чином, щоб максимальна оцінка отримала число 1, а число  $n$  - мінімальне. Оскільки всі  $n$  рейтингів різні, то відповідні числа в натуральному ряду є оцінками оцінок першого експерта. Якщо рейтинги  $i$ -го експерта збігаються, то цими оцінками присвоюється такий же ранг, який дорівнює середньому арифметичному значенням відповідних чисел цілого числа. Якщо експерт вважає себе не компетентним у тих чи інших питаннях і не оцінює той чи інший фактор, то при ранжируванні суми оцінки даного чинника береться за середнє статистичне значення оцінки ( $M_j$ ) цього фактора.

Сума рангів визначається за формулою

$$S_j = \sum_{i=1}^{m_1} R_{ij},$$

де  $R_{ij}$  – ранг оцінки;

$m_1$  – кількість людей, що оцінили один фактор (частіше  $m_1 = m$ ).

При порівнянні важливості будь-яких факторів за  $S_j$  найбільш важливим слід вважати той, що характеризується найменшим значенням  $S_j$ .

Поряд з показниками відносної важливості, необхідно визначити ступінь узгодженості думок експертів. Ступінь згоди експертів щодо важливості  $j$ -фактора визначається коефіцієнтом варіації

$$V_j = \frac{\sigma_j}{M_j},$$

де  $\sigma_j = \sqrt{\frac{1}{m_j - 1} \sum_{i=1}^{m_j} (C_{ij} - M_j)^2}$  – оцінка відхилення ( $m_j$  – кількість експ., які оцінили  $j$ -й факт.).

Коефіцієнт варіації ( $V$ ) визначається для кожного фактора і описує ступінь узгодженості експертних висновків до відносної важливості  $j$  фактора.

Ступінь узгодженості експертних висновків до відносної важливості сукупності всіх пропозицій по оцінці факторів визначається коефіцієнтом відповідності

$$k = \frac{\sum_{j=1}^n d_j}{m_1^2 * (n^3 - n) - m_1 \sum_{i=1}^{m_1} T_i},$$

де  $m_1$  – к-сть експертів, що оцінили;

$n$  – к-сть факторів;

$d_j$  – відхилення суми рангу оцінок, отриманих  $j$ -м фактором, від середньої арифметичної суми рангу оцінок всіх чинників;

$T_i$  – це індекс пов'язаних (рівних) оцінок, призначених  $i$ -м експертом.

Відхилення  $d_j$  визначається за формулою

$$d_j = S_j - M[S_j],$$

де  $M[S_j] = \frac{\sum_{j=1}^n S_j}{n}$  – середнє арифметичне сум рангів, що оцінили.

Якщо все  $n$  оцінок, визначених  $i$ -м експертом, різні, то  $T_i = 0$ ; якщо оцінки однакові, то

$$T_i = \sum_{l=1}^{\alpha} (t_l^3 - t_l)$$

де  $\alpha$  – к-сть груп зв'язаних рангів;

$t_l$  – к-сть зв'язаних рангів в  $l$  групі.

Коефіцієнт конкордативності може варіюватися від 0 до 1 і визначається для кожного питання типу "оцінка відносної важливості". У разі повної узгодженості експертних висновків  $k = 1$ . Якщо значення коефіцієнта узгодження невелика, це означає, що серед експертів існує слабкий консенсус. Причиною низького консенсусу серед експертів може бути загальна думка експертів, а може і ні, або в групі експертів з високим ступенем консенсусу, але їх загальні думки суперечать один одному.

Для того щоб визначити групу експертів, серед яких спостерігається висока ступінь узгодженості думок, пропонується виключити одного експерта в сукупності і визначити коефіцієнт відповідності для інших експертів. Якщо коефіцієнт відповідності в цьому випадку збільшився, то цей експерт виключається в сукупності. Якщо він став менше, то цей експерт залишається в сукупності. Такі розрахунки проводяться послідовно для кожного експерта. В результаті підвищується ступінь координації між експертами, які залишаються в сукупності, збільшується.

Рівень активності експертів визначається співвідношенням

$$k_{aj} = \frac{m_j}{m}$$

де  $m_j$  – к-ть експертів, які оцінили фактор;

$m$  – к-ть експертів, які приймали участь в оцінюванні.

Чим вище рівень активності експертів, тим більше вони вважають себе

компетентними в оцінці j-фактора.

У практиці військового прогнозування типовими завданнями є визначення значення певного параметру або набору параметрів, характерних для даного об'єкта прогнозування. Розглянемо деякі особливості прогнозування значень параметрів методом масштабних оцінок.

Якщо кожен з  $m$  експертів, що беруть участь в опитуванні, дає одне значення параметру  $C_i$ , то результати дослідження даних можуть дати наступні характеристики, що оцінюють як майбутню вартість параметра, так і ступінь координації дій експертів щодо даної оцінки.

Середнє значення експертних оцінок (точковий прогноз цієї групи експертів), що характеризує загальну думку експертів:

$$\tilde{C} = \frac{1}{m} \sum_{i=1}^m C_i .$$

Середньоквадратичне відхилення, що характеризує розкид думок (тактичний прогноз) окремих експертів щодо середнього значення:

$$\tilde{\sigma}_c = \sqrt{\frac{1}{m-1} \sum_{i=1}^m (C_i - \tilde{C})^2} .$$

Коефіцієнт варіації, який характеризує ступінь однотайності експертів:

$$V = \frac{\tilde{\sigma}_c}{\tilde{C}} .$$

Данні показники також дозволяють прогнозувати розмір області (довірчого інтервалу), яка із заданою вірогідністю досягає значення прогнозованого значення.

Майбутня вартість прогнозованого значення визначається наступним чином

$$\tilde{C} - \varepsilon_1 \leq C \leq \tilde{C} + \varepsilon_2 .$$

Для визначення цього регіону необхідно зробити припущення про тип розподілу величини величини  $C_i$ . Коли велике число експертів цей розподіл можна вважати нормальним. Для  $m < 10$  можна використовувати дистрибутив Student. Тому що обидва розподілу симетричні, то

$$\varepsilon_1 = \varepsilon_2 = t_\beta \frac{\tilde{\sigma}_c}{\sqrt{m}} ,$$

де  $t_{\beta}$  – визначається за правилами математичної статистики.

Метод парних порівнянь. Серед методів експертної оцінки, застосованих для одержання коефіцієнтів відносної важливості факторів (параметрів, ознак, напрямку розвитку і т. ін.), метод парних порівнянь вважається дуже ефективним, тому що дозволяє визначити відносну важливість факторів, коли безпосереднє порівняння стає важким.

Згідно з цим методом усі фактори порівнюються між собою послідовно, причому кожна наступна оцінка не зв'язана з попередньою.

Для полегшення процедури порівняння звичайно використовують спеціальну таблицю матрицю парних порівнянь (таблиця. 1.3), в якій фактори (параметри, ознаки, напрями розвитку) розміщені за горизонталлями та за вертикалями (у верхньому рядку та в лівому крайньому стовпці).

Таблиця 1.3. Матриця парних порівнянь

Фактори	$y_1$	$y_2$	...	$y_j$	...	$y_n$
$y_1$	$y_1 : y_1$	$y_1 : y_2$	...	$y_1 : y_j$	...	$y_1 : y_n$
$y_2$	$y_2 : y_1$	$y_2 : y_2$	...	$y_2 : y_j$	...	$y_2 : y_n$
...	...	...	...	...	...	...
$y_i$	$y_i : y_1$	$y_i : y_2$	...	$y_i : y_j$	...	$y_i : y_n$
...	...	...	...	...	...	...
$y_n$	$y_n : y_1$	$y_n : y_2$	...	$y_n : y_j$	...	$y_n : y_n$

Для того щоб уніфікувати відповіді експертів та зробити більш простою початкову обробку даних, всім експертам повинен бути заданий однаковий спосіб оцінки (або у вигляді числа, або у вигляді відношення чисел) і одна шкала оцінок. Оцінка не повинна бути більша базового числа. Шкала оцінок не повинна бути дуже великою, але не дуже дрібною (перше знижує суворість оцінки, друге ускладнює обчислення).

Метод Дельфи. Одним із перспективних методів прогнозування є метод Дельфи. В методі Дельфи замість колективного обговорення проблем проводиться індивідуальне опитування експертів звичайно в формі анкет для обчислення відносної важливості і термінів здійснення подій.

В основу методу Дельфи покладені такі положення:

питання, що ставляться перед експертами, повинні допускати можливість відповіді у вигляді числа;

експерти повинні мати достатньо інформації для того, щоб дати оцінку;

відповідь на кожне питання (оцінка) повинна обґрунтовуватися експертом.

Процедура проведення опитування проводиться в чотири тури. Під час кожного туру експерт повинен висловлювати свою думку у вигляді числа за заздалегідь підготовленою шкалою оцінок.

Під час першого туру експерт повинен дати оцінку подіям, перелік яких розроблений аналітиком. Після першого туру опитувань аналітики проводять статистичну обробку отриманих оцінок, уточнюють перелік подій та аналізують оцінки. Припустимо, що від експертів отримали деяке число оцінок. Ці оцінки упорядковуються, наприклад, у порядку зменшення. Визначається медіана, за яку беруть середній член ряду, по відношенню до якого число оцінок з початку та з кінця ряду буде однаковим.

Діапазони цих кватилів у першому наближенні дорівнює значенням оцінок ряду в інтервалі, який дорівнює 25% від початку і 25% від кінця ряду.

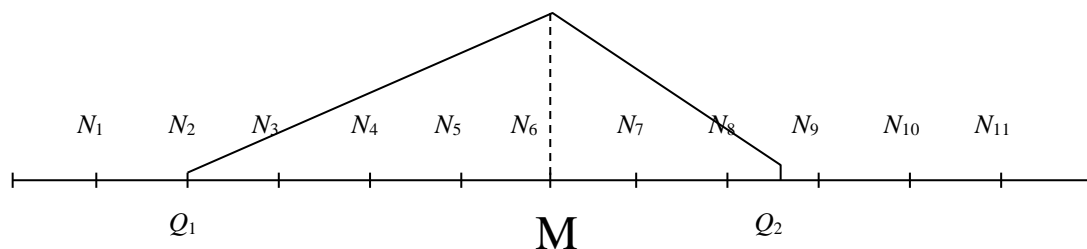


Рисунок. 1.4. Медіана та кватилі в методі Дельфи

Таким чином, медіана і кватилі створюють на осі ряду чотири інтервали, серед яких двом середнім  $Q_1M$  і  $Q_2M$  віддається перевага.(рисунок 1.4).



Показники, що отримані таким чином, беруться за характеристики розподілу оцінок: медіана є характеристикою групової відповіді, а діапазон кuartилів  $Q_1M$  і  $Q_3M$  – показниками розкидання індивідуальних оцінок.

Після першого туру кожному експертові повідомляється значення медіани і розмах між крайніми кuartилями. Експертів, оцінки яких попали в крайні кuartилі, просять обґрунтувати, тобто пояснити причини розходження з груповою думкою. Експерти можуть наводити будь-які аргументи (такі ж самі, які б приводили під час дискусії). Різниця полягає тільки в тому, що ці аргументи анонімні. Експерти можуть переглянути свою думку і при бажанні виправити оцінки. З отриманим обґрунтуванням знайомлять решту експертів, не визначаючи при цьому чиї вони. Така процедура дозволяє всім експертам враховувати обставини, які вони могли випадково не помітити, або якими зневажили в першому турі опитування.

Потім проводиться другий тур, здійснюється статистична обробка оцінок, з результатами якої знову знайомлять експертів. Всі експерти під час другого туру повинні обґрунтовувати свої оцінки, врахувати заперечення і прокоментувати їх.

Після третього туру опитування і відповідної статистичної обробки оцінок експертів, оцінки яких знову не ввійшли в інтервал між крайніми кuartилями, просять надати контраргументи на користь своїх оцінок. На четвертому етапі експертизи всі обґрунтування знову доводяться до експертів, які мають останню можливість скорегувати свої відповіді. Медіана цих остаточних відповідей і береться за оцінку, найбільш близьку до єдиної думки групи експертів. Якщо експерти приходять до згоди раніше, наприклад після другого туру, то проводити всі чотири тури недоцільно.

Практика показує, що основні результати використання методу Дельфи полягають у наступному. Типовим для першого туру опитування є широке розкидання індивідуальних оцінок. За допомогою здійснення ітерацій і оберненого зв'язку індивідуальні відповіді сходяться, а групова відповідь (медіана остаточних індивідуальних оцінок) стає точнішою.

Недоліками методу є необхідність значних витрат часу на процедуру експертизи, а також неможливість прямого обговорення думок експертів.

## 2.2 Аналіз наявних інструментів

ПЗ для прогнозування попиту і планування ланцюга поставок використовується в різних галузях і на підприємствах усіх розмірів. Ринок для цього виду програм постійно зростає, так як технології планування і прогнозування продовжують розвиватися. Деякі продавці пропонують в комплекті з прогнозами також і софт для планування ланцюга поставок з різними функціями, такими як контроль і планування поповнення запасів, інші ж пропонують планування продажів як доповнення або самостійної функції.

Програмне забезпечення для прогнозування дозволяє організаціям мінімізувати витрати, Відстежуючи такі параметри як тренди, рівні і сезонність, які можуть впливати на майбутній попит. Ці типи систем виконують завдання підвищення ефективності управління прогнозуванням для того, щоб усунути помилки або похибки в даних, і навіть дозволяють отримувати дані попиту в режимі реального часу.

Особливості софта для прогнозування попиту

Історичний аналіз. Крос-функціональні або багатовимірні процеси прогнозування аналізують "історичний" попит на окремий продукт, а також беруть до уваги більш широкі макро - і мікро-економічні тенденції, щоб запропонувати більш точний прогноз керівникам управління ланцюгами поставок.

Поділ даних. Програмне забезпечення вміє збирати широкий спектр даних для поліпшення прогнозування попиту. Сегментація по продукту, знаходження сезону та іншої корисної інформації при формуванні прогнозу - допомагають знизити помилки.

"What if" аналіз. Маючи широкий діапазон можливостей, програма може виконувати кілька сценаріїв і моделей, заснованих на підході "що-якщо". Грунтуючись на результатах цих "що-якщо" моделей, можливі відхилення від планованого попиту можуть бути прив'язані до окремих ланках ланцюга поставок для коригування відвантажень або виробництва.

Комунікація в ланцюзі постачань. Софт для прогнозування взаємодії з іншими елементами SCM систем, переглядаючи графіки поповнення запасів, враховуючи наявні можливості. Це запобігає перевиробництву і додаткові витрати, такі як витрати на зберігання, інвентаризацію та обробку запасів, які не потрібні найближчим часом.

Програмне забезпечення як послуга (SaaS). Багато компаній вибирають хмарні рішення, коли не несучи серйозних капітальних витрат, можна швидко "розгорнути" програму через Інтернет. У такого підходу є свої плюси і мінуси, про які ми ще поговоримо, але варто відзначити, що хмарні рішення, поряд з «ліцензіями» вже пропонують і лідери ринку SAP, S2K і Infor.

Цифровий моніторинг. З ростом глобалізації ланцюжків поставок, компанії все частіше стикаються з такими проблемами, як затримки поставок, збільшення термінів доставки (невизначений lead time), а також зниження якості товарів і зростання цін. Для зниження впливу цих проблем і збереження прозорості та контролю по всьому ланцюжку поставок, багато програм для планування і прогнозування пропонують функцію «цифрового» контролю, яка забезпечує оновлення важливою для вас інформацією в реальному часі.

Штучний інтелект. Ланцюжка поставок для глобальних компаній-це складний і постійно що розвивається, що означає, що він сприйнятливий до зовнішніх проблем, які можуть коштувати мільйони, якщо не мільярди доларів. Для боротьби з цим деякі виробники розробляють нові функції, які здатні знаходити і виправляти історичні та поточні помилки в ланцюжку поставок. Наприклад, Rulex представила програмне рішення II у вересні 2016 року, яке може автоматично виявляти і виправляти помилки в даних ланцюжки поставок.

Підвищений попит на сторонню інтеграцію. Замість того, щоб купувати повністю нову систему, яка охоплює всі аспекти бізнес-операцій, багато компаній шукають рішення, які дозволять їм легко імпортувати дані, такі як закупівлі, продажу і операції, з їх існуючого програмного забезпечення в нове рішення без додаткових трудовитрат.

Аналітика. Розробники софта використовують різні моделі прогнозування для

підвищення точності. Як результат, в область "софт для прогнозування" приходять компанії з бізнес-аналітики (business intelligence) і пропонують софт прогнозування попиту і планування для застосування в логістичному ланцюжку. Наприклад, Qlik і Border BI розширили свої компетенції в плануванні і прогнозуванні.

Вигоди від софту для прогнозування попиту і планування

Софт для прогнозування попиту і планування може принести чимало вигод для організації. Ці переваги включають в себе (але не обмежуються):

Підвищення ефективності. Софт для планування попиту і прогнозування дозволяє підвищити ефективність ланцюжка поставок компанії і операцій, поліпшити продуктивність і контролювати запаси. Цей інструмент може також створювати план попиту на основі історичних даних і прогнозувати попит в майбутньому.

Поліпшення співпраці. Софт для планування може допомогти поліпшити зв'язок між ланками ланцюжка поставок компанії та іншими підрозділами, включаючи продажі, маркетинг та фінанси. Це може допомогти зі створенням плану дій для різних сценаріїв.

Контроль витрат. Точний прогноз попиту дозволяє знизити витрати, скорочуючи оборотний капітал, тобто складські запаси. Ця функція запобігає затоварення, фокусуючись на високоприбуткових товарах і їх оборотності.

Аналіз тренду. На додаток до контролю над витратами, аналіз тренду, рівня і сезонності допоможе згенерувати прогнози на підставі історичних даних і поєднуючи їх з даними, пов'язаними з сезонними коливаннями і акціями (розпродажі в Новий рік?). Ці прогнози можуть допомогти запобігти ситуація, коли попит є, а товару немає.

## LLAMASOFT DEMAND GURU

LLamasoft Demand Guru - флагманський продукт від компанії LLamasoft для прогнозування і сценарного аналізу попиту.

LLamasoft Demand Guru допомагає виявляти і вивчати ключові фактори попиту, прогнозувати сценарії попиту на товари, отримувати достовірну інформацію для прийняття впевнених стратегічних рішень з управління ланцюгами поставок.

## Можливості LLamasoft Demand Guru

LLamasoft Demand Guru дозволяє:

Вивчати ключові фактори попиту, використовуючи сценарне моделювання та інтерактивну візуалізацію, а також легко декомпонувати свій попит на сезонність, тренд, вплив бізнесу і зовнішніх аномалій.

Мати кілька моделей для прогнозування одних і тих же рядів - для кращого розуміння характеристик попиту і їх аналізу.

Будувати прогнозні моделі з використанням сучасних алгоритмів машинного навчання, при цьому не потрібно глибокого розуміння принципів їх роботи.

Проводити аналіз сезонності попиту, даних про тренд і зовнішніх факторах для швидкого і точного прогнозування майбутнього ринкового попиту на короткострокові, середньострокові і довгострокові горизонти.

Отримати доступ до Data Cube - постійно оновлюється сховища даних про погоду та економіці. Використовуйте ці дані в якості факторів для поліпшення ефективності прогнозування.

Інтегруватися з будь-якими існуючими корпоративними системами, завдяки вбудованому ETL модулю. Безшовна інтеграція Demand Guru з іншими продуктами від LLamasoft дозволить управляти попитом і ланцюжками поставок ще ефективніше.

Переваги LLamasoft Demand Guru:

- Зручний, інтуїтивно зрозумілий інтерфейс.
- Наявність ETL-модуля для попередньої обробки та очищення вхідних даних.
- Data Cube - вбудований, регулярно оновлюється сервіс для надання макроекономічних показників по регіонах. При прогнозуванні враховуються такі чинники, як населення, погода або купівельна спроможність.
- Використовує передові алгоритми прогнозування.
- Можливість додавання власних алгоритмів прогнозування.
- Інтерфейс сценарного аналізу дозволить отримати в зручному форматі

кілька варіантів прогнозу, в залежності, наприклад, від величини промобюджету.

## KORUS | FORECAST

KORUS | Forecast - хмарний сервіс для прогнозування попиту, розроблений фахівцями «КОРУС Консалтинг» з урахуванням галузевої специфіки ритейлу.

KORUS | Forecast автоматично робить аналіз даних і розраховує прогноз майбутнього попиту на товари, мінімізує вплив людського фактора і ризику можливих помилок.

### Переваги KORUS | Forecast:

- Готове масштабується, для прогнозування попиту.
- Швидке підключення і гарантовану якість сервісу.
- Логістична аналітика в режимі реального часу.
- Можливість оцінки рівня сервісу і ефективність маркетингових акцій.
- Робота з даними будь-якого формату.
- Консультаційна підтримка.

KORUS | Forecast не вимагає дорогого впровадження і покупки додаткового ІТ-обладнання. Всі обчислення проводяться в хмарі, а ви отримуєте готові звіти і прогнози для прийняття рішень.

### Можливості KORUS | Forecast

Підключаючись до сервісу KORUS | Forecast, ви отримуєте набір налаштованих аналітичних звітів, які дозволяють відслідковувати в режимі реального часу найбільш важливі показники за прогнозом і роботі логістики.

Звіт з оцінки прогнозу на будь-якому рівні деталізації географічної і товарної ієрархії.

Звіт, що показує помилки прогнозування і причини їх виникнення.

Звіт про рівень сервісу логістики по всій мережі з можливістю деталізації до конкретного магазину (може бути доопрацьований під ваші вимоги).

Звіт по періоду оборотності товарів з агрегацією до тижня, що дозволяє отримати інформацію в динаміці.

Звіт по списанню продукції.

Звіт по ефективності промо-акцій.

Звіт по неліквідних запасах.

Діаграма розсіювання по відсутності товару на полиці в розрізі категорій, магазинів.

Діаграма розсіювання по оборотності і товарному запасу.

Підключення до KORUS | Forecast

Підключення, настройка сервісу і автоматичного завантаження даних із зовнішніх ІТ-систем займе сім тижнів. Ви отримуєте готове рішення для прогнозування попиту і пакет консультаційної підтримки.

### **3. Реалізація задачі прогнозування на основі теорії фракталів**

При розробці програмного забезпечення для кінцевого користувача, головний акцент було зроблено на простоті, точності та зручності під час користування програмним забезпеченням.

#### ***3.1 Поняття теорії фракталів для задачі прогнозування***

Вивчення властивостей великомасштабних часових рядів (самоподібність, періодичність, нестационарність, аномальне розподіл) і аналіз існуючих методів, і моделей показали складність роботи з ними і відсутність універсального підходу до процесів розпізнавання і відтворення. Крім того, відомі моделі мають недоліки: значна складність аналітич виразів, однопараметричними, неактивний, складність інтерпретації поза галузями, де була використана модель.

Відокремлений клас становлять моделі фрактальних функцій (Больцанно, Ріман, Вейерштрасс і ін.), які мають велике перспективне значення для вирішення завдань в радіоелектроніці, механіці, теорії хаосу і ін. Проте, оскільки вони представляють собою набір розрізнених функцій, деякі навіть не мають аналітичного погляду, між ними немає ніякого взаємозв'язку, все це ускладнює їх практичне застосування. Тому перед нами стоїть завдання створення універсальної моделі, яка дозволила б замінити або повторити відоме.

Аналіз властивостей фракталів і сплайнів і їх застосування для обробки часових рядів показав перспективність модифікації моделі класичного ермітського сплайна шляхом додання їй фрактальних властивостей для використання при аналізі великомасштабних часових рядів. Підставою для цієї тези є наступні пункти:

1) Сплайн і фрактали використовуються в комп'ютерній графіці для моделювання складних кривих: гладких технічних (сплайн) і нерівномірних природних (фрактал).

2) Збільшення числа робіт, в яких одне й те саме явище або процес (частота серцевих скорочень, берегова лінія, фінанси) досліджується сплайнами або



фракталами.

3) Близькість властивостей: безперервність, самоподоба, негнучкість, недифференційованості (останні два пункти - тільки для зламаних).

Таким чином, інкапсуляція сплайнів і фрактальних властивостей в одному об'єкті є можливим і доцільним кроком для аналізу великомасштабних часових рядів.

Розробка нового класу сплайнів моделей ставить задачу визначення потужності сплайнів нових типів і узагальнення методів сплайнової інтерполяції, апроксимації та оптимізації для даного класу моделей. Практичним завданням в цьому контексті є розробка алгоритмів розпізнавання, відтворення і прогнозування великомасштабних часових рядів з використанням фрактальних сплайнів.

Аналіз програмних засобів дозволив виявити наступні недоліки: підвищена увага до обробки двовимірних зображень, обмежений набір виконуваних операцій, відсутність коштів для поліпшення результатів розпізнавання, складний інтерфейс. Виходячи з цього, ставиться завдання розробки бібліотеки функцій, яка дозволяє автоматизувати розрахунок пропонованої моделі, і програми на її основі, яка не має наступних недоліків.

Визначення. *Фрактальний сплайн* – неперервна і визначена на фрагментах функція  $^fk S(t)$ , що складаються з самоподібних сегментів, які є функціями одного типу шкали  $k$  і пов'язані особливим чином. В якості генераторів функцій обрані ергономічні сплайни: 1) зберегти похідні значення в точках стику; 2) не перевищувати значення табличній функції на відміну від природного сплайна, що важливо для задач інтерполяції; 3) є локальними; 4) легко обчислюються.

Самоподібність досягається рекурсивним методом побудови фрактального сплайна. Материнський сплайн (сплайн нульовий шкали) побудований у вигляді лінійного або кубічного сплайна Ерміта. Далі кожен з його фрагментів замінюється зменшеною копією батьківського сплайна і додається до сплайну попереднього масштабу. Процес повторюється на необхідну глибину. У загальному випадку немає обмежень по довжині і частці розміщення фрагментів. Сплин періодичності забезпечує гладку стиковку вкладених копій. Схематично процес побудови

фракційного сплайна першого ступеня представлений таким чином на рисунку 2.1.

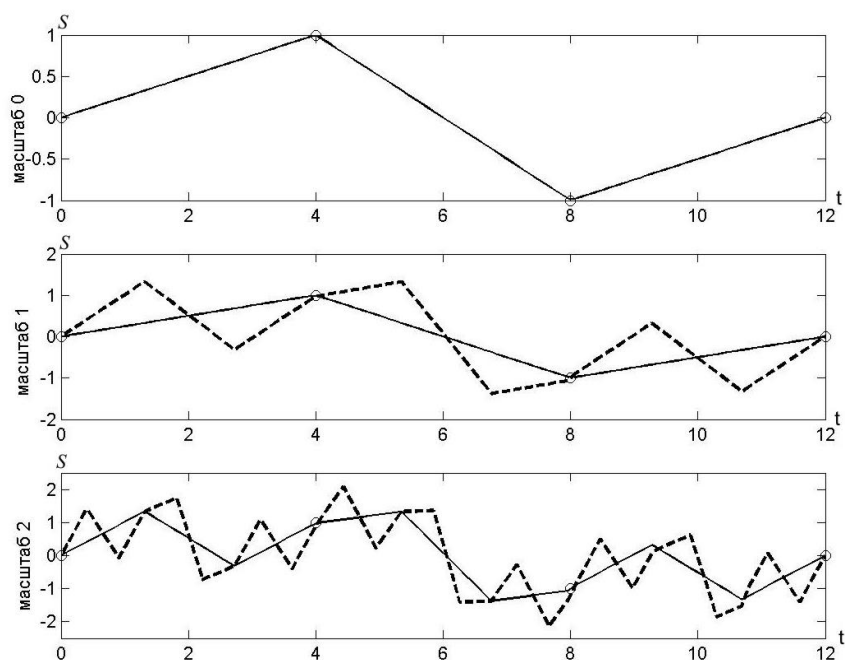


Рисунок 2.1 – Процес побудови факталу та сплайну

Розглянемо побудову фрактального сплайну повторюваним методом на повторно рівномірної сітці вузлів для деякого масштабу  $k$ .

Маємо ермітів сплайн  $S(t)$ , що на інтервалі  $t \in [a; b]$  задається поліномом

$$S(t) = \sum_{j=0}^R a_j B_j(t),$$

де  $a_j$  – числові коефіцієнти, рівні значенню сплайна у вузлах,  $B_j(t)$  – базисні ермітові сплайни,  $R$  – кількість фрагментів сплайна.

Приймемо його за материнський сплайн нульового масштабу  ${}^0S(t) = S(t)$ .

Побудуємо сплайн 1 масштабу фрагментано. Для цього треба замінити кожен фрагмент сплайну нульового масштабу його зменшеною реплікою:

$${}^1S_j(t) = {}^0S(m_j t),$$

де  $j$  – номер фрагменту,  $m_j$  – множник, який дорівнює співвідношенню довжини  $j$ -го фрагмента до довжини сплайну:

$$m_j = \frac{t_j - t_{j-1}}{b - a}.$$

Якщо вузли сплайна розміщені рівномірно, то  $m_1 = m_2 = \dots = m_R = m$ .

Фрактальний сплайн 1-го масштабу є сумою сплайна 0-го масштабу та сплайна 1-го масштабу на всіх фрагментах:

$${}^1f S(t) = {}^0S(t) + \bigcup_{j=1}^R {}^1S_j(t) \quad (2.1)$$

Але послідовність сплайнів також є важливим сплайном, тоді праву частину рівняння можна виразити так:

$$\bigcup_{j=1}^R {}^1S_j(t) = \bigcup_{j=1}^R {}^0S(mt) = \bigcup_{j=1}^R \sum_{j=0}^R a_j B_j(mt) \quad (2.2)$$

Оскільки сплайнові значення першої шкали в вузлах зберігаються, то, незважаючи на те, що кількість фрагментів збільшується в R разів, кількість параметрів залишається незмінним. Замістити (2.2) у (2.1), звівши подібні доданки та виконавши заміну  $B'_j(t) = B_j(t) + B_j(mt)$ , остаточно маємо:

$${}^1f S(t) = \sum_{j=0}^R a_j B'_j(t) \quad (2.3)$$

Для другої шкали процес повторюється, тільки масштабний коефіцієнт множиться. Так як це залежить від ширини сплайнів фрагментів, і вони пропорційні кожній k-ой шкалою, то ми маємо наступне:

$$m_{jk} = (m_j)^k.$$

Основні функції для фрактального сплайна другого масштабу мають вигляд:

$$B''(t) = B'_j(t) + B'_j(mt) = B_j(t) + B_j(mt) + B_j(m^2t).$$

Фрактальний сплайн другого масштабу розраховується як:

$${}^2f S(t) = \sum_{j=0}^R a_j B''_j(t) = \sum_{j=0}^R a_j \sum_{k=0}^2 B_j(m^k t).$$

Процес можна продовжити, взявши за основу (2.2)-(2.3), й мати рекурентні формули. Узагальнемо фрактальний сплайн довільного масштабу K визначається

так:

$${}^{kf}S(t) = \sum_{j=0}^R a_j \sum_{k=0}^K B_j(m^k t), \quad (2.4)$$

У матричному вигляді рівняння (2.4) мають вигляд:

$${}^KS = {}^0S + {}^1S + \dots + {}^KS = {}^0PA + {}^1PA + \dots + {}^KPA = ({}^0P + {}^1P + \dots + {}^KP)A = {}^KPA.$$

Тобто фрактальний сплайн може бути представлений як добуток фрактальної матриці планування і матриці коефіцієнтів, причому для всіх масштабів.

Фрактальна матриця планування є самоподобной, тобто матрична структура  ${}^0P$  повторюється  $k$  раз у структурі матриць  ${}^1P, {}^2P, \dots, {}^KP$  в маленькому масштабі (рисунок 2.2). Останні в силу локальних властивостей функції форми є блочно-діагональними.

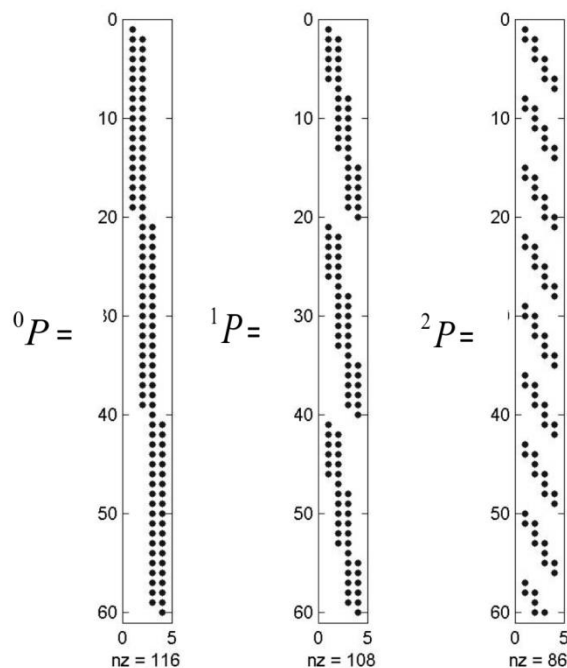


Рисунок 2.2 – Внутрішня структура матриць

Значення  $j$ -го стовпця матриці є значеннями функції  $j$ -го виду для  $t$ .

Формальна функція кубічного фрактального сплайна і його похідних показана на рисунку 2.3. Перша похідна неперервна і складається з  $4k$  фрагментів, які представляють собою фрактальні параболи. Друга похідна має розриви і складається з  $4k$  прямих ліній.

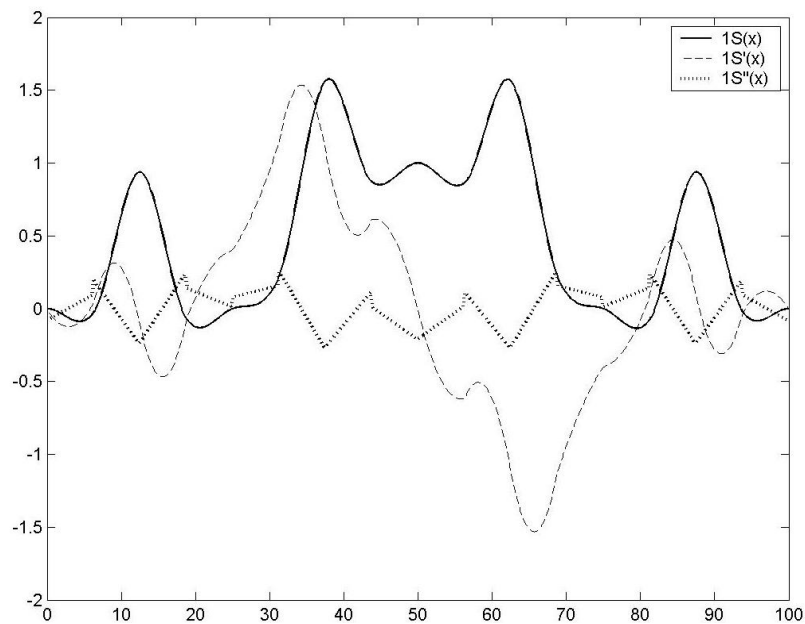


Рисунок 2.3 – Функція форми кубічного сплайна першого масштабу (суцільна лінія) та її похідні: перша (штрихова) та друга (пунктирна)

Головними етапами розробленого методу побудови фрактального сплайна є: вибір вузлів сітки і граничного масштабу; формування матриці планування сплайна на кожному масштабі; підсумовування часткових матриць і отримання фрактальної матриці; розрахунок фрактального сплайну.

На рисунку 2.4 наведено приклад використання повторюваного методу побудови лінійного фрактального сплайна другої шкали на дискретній множині точок, батьківський сплайн яких має 5 рівновіддалених вершин.

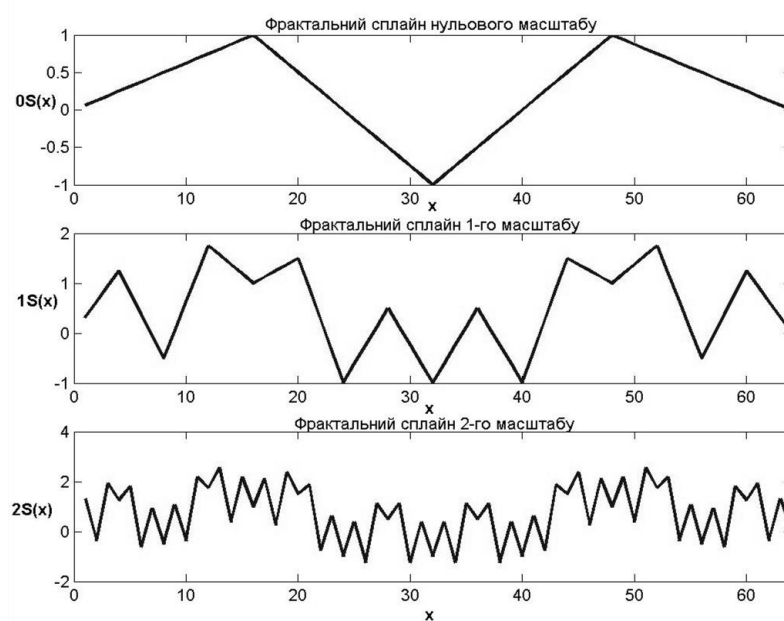


Рисунок 2.4 – Лінійний сплайн

Доведено, що лінійний фрактальний сплайн з безліччю масових скупчень має властивості, аналогічними властивостями фрактальних функцій, а саме:

- 1) зближення і спадкоємність;
- 2) відсутність похідного фінансового інструменту в будь-який момент часу;
- 3) дробове вимір Хаусдорфа-Бесіковича - більше одного.

Важливою особливістю даної моделі є те, що в залежності від розташування сплайна материнської плати можна отримати криві, повністю збігаються з деякими відомими фракталами і фрактальні функціями: крива Коха, функції Такагі, Больцано, Вейерстрасса (рисунок 2.5) тощо. Це свідчить про універсальність моделі.

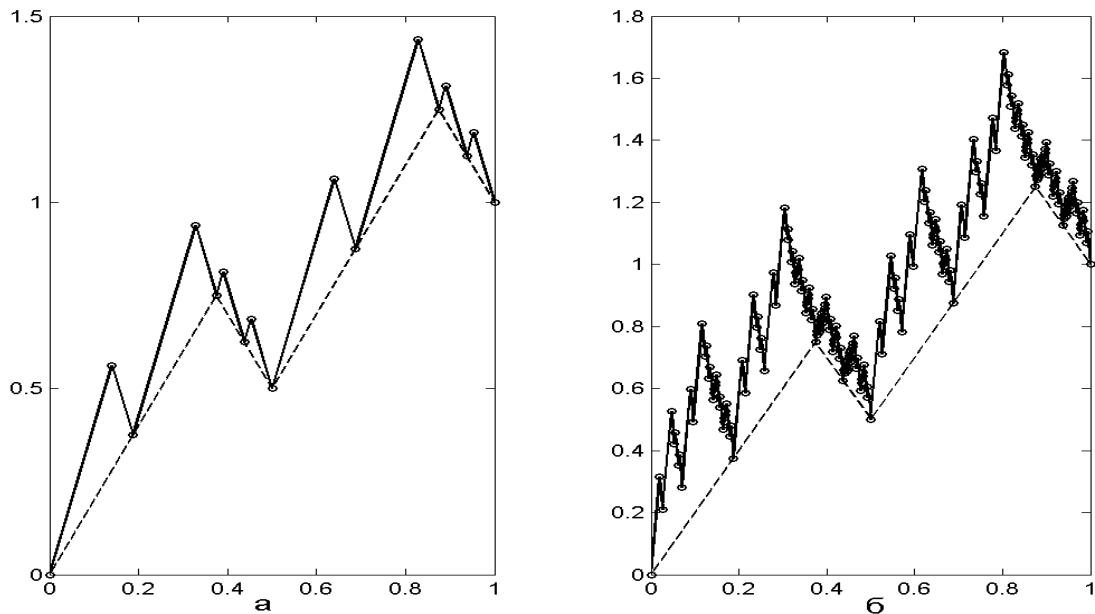


Рисунок 2.5 – Функція Больцано і фрактальний сплайн  $k$ : а)  $k=2$ , б)  $k=4$

Разовий метод побудови фрактального сплайна полягає в відборі рядків з матриці планування  ${}^0P$  для отримання матриці  ${}^kP$  за правилом:

$$z = 1 + \frac{q-n}{N-n}(j-1),$$

де  $z=1..q_k$  – номер рядка матриці  ${}^kP$ ,  $j=1..N$  – номер рядка матриці  ${}^0P$ .

Зміна параметрів фрактального сплайна (зміна ширини фрагментів, кількості лусочок, частки вкладених сплайнів, додавання випадкових компонент і т.д.) дозволяє створювати різноманітні масштабні криві і прозоро контролювати їх форму.

Від реалізації методу побудови фрактального сплайна залежить:

- Періодичність базису (періодична, майже періодична, неперіодичних);
- розміщення вузлів на сітці (рівномірне, нерівномірне);
- коефіцієнт подібності сплайнів на фрагментах (однаковий для всіх фрагментів, зміни пропорційно ширині фрагмента, що задається переглядом користувачем, вибирається випадково);
- повторення базису на шкалі (одна і та ж основа для всіх фрагментів на всіх шкалах, базис залежить від шкали, базис залежить від кількості фрагментів).

Пропонується розглянути такі фрактальні сплайни, як кардинали, для яких

виконані умови періодичності, однорідності, однакового коефіцієнта подібності і єдиної основи. Порушення хоча б однієї з умов робить фрактал сплайна не серцевим, а саме: неперіодичним (між фрагментами є зазори на інвестованих масштабах), нерівномірним (складна конструкція, можливо, нерівномірні масштаби на фрагментах), самостверджуватися (з різними масштабними факторами) або мультифрактального (побудованим на декількох підставах) (рис.2.6). У той же час, такі сплайни більше відповідають нерегулярним кривим природного походження і, отже, мають великі перспективи реалізації, але в той же час мають ускладнення моделі.

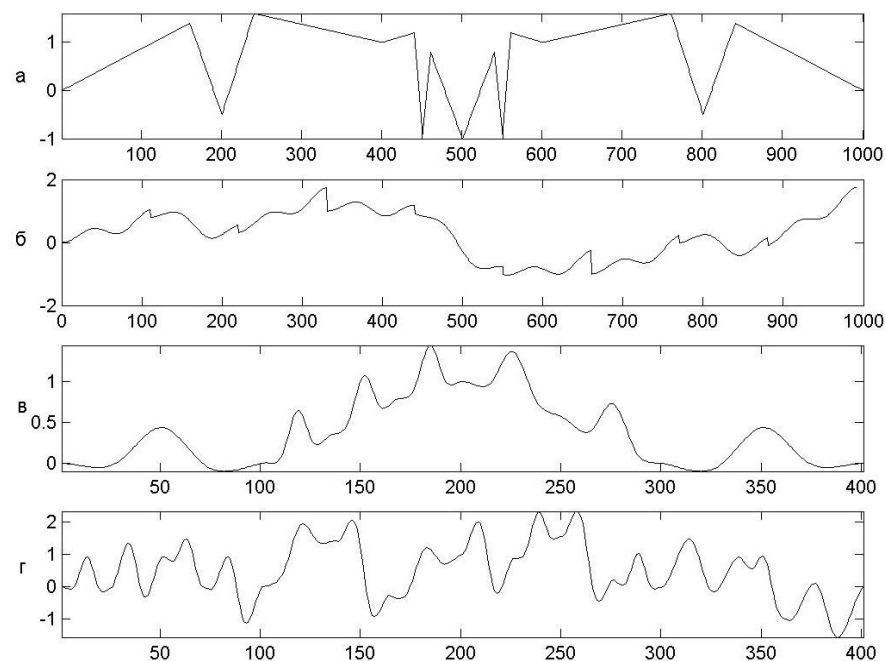


Рисунок 2.6 – Некардинальні сплайни: а) самоафінний; б) неперіодичний; в) нерівномірний; г) мультифрактальний

### ***3.2 Розробка методики використання теорії фракталів в задачі прогнозування***

Інтерполяція фрактальними сплайнами відноситься до класу методів фрактальної інтерполяції і має такі переваги:



- 1) дуже складні функції однозначно визначаються невеликою кількістю параметрів;
- 2) параметри функції екстраполюються від однієї шкали до іншої, що дозволяє деталізувати функцію зі збереженням її властивостей;
- 3) можлива інтерполяція негладких і повсюдно недиференційованих функцій;
- 4) дозволяє працювати з нерівномірно розподіленими вузлами інтерполяції.

Алгоритм фрактальної сплайнової інтерполяції включає в себе наступні кроки:

1. Вкажіть або отримаєте: набір точок інтерполяції, тип і довжина базової сплайни, максимальний масштаб  $k_{\max}$ .
2. Формувати матрицю вузлів сплайна нульового масштабу.
3. Розрахувати матрицю планування для сплайна нульового масштабу.
4. Повторіть дії на стор. 1-3 для шкал від першого до  $k_{\max}$ .
5. Додати матриці планування з пункту 4 до числа рядків в таблиці з пункту 2 шляхом дублювання рядків.
6. Додайте матрицю планування всіх шкал, починаючи з нуля.
7. Розрахувати фрактальну матриці планування (п.6) і матриці коефіцієнтів (ординат вузлів).

### ***3.3 Побудова алгоритму прогнозування на основі теорії фракталів***

На рисунку 2.7 показаний приклад роботи алгоритму 10 пунктів: (0,0), (10, 1), (20, 0.25), (30, 0.5), (40, -0.5), (50, 0.75), (60, 0), (70, 0.25), (80, 0.8), (90, 0). Інтерполяція здійснюється кубічним фрактальним сплайном з поступовим поглибленням шкали від нуля до третини. Пунктирна лінія позначена основним сплайном - генератором фракталів.

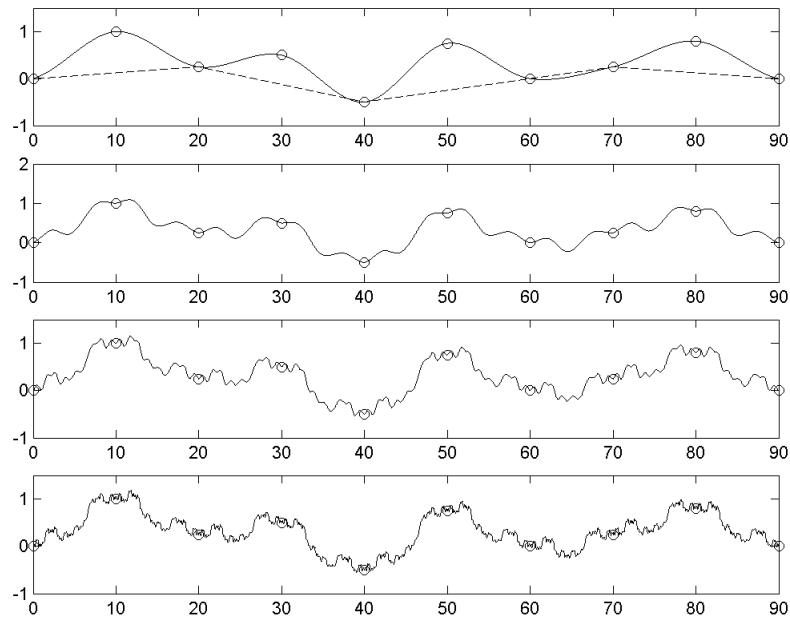


Рисунок 2.7 – Інтерполяція фрактальних сплайнів

Апроксимація фрактальним сплайном. Оцінка параметрів фрактального сплайна може розглядатися як завдання обернено пропорційна інтерполяції. У ньому необхідно, щоб відомі значення в точках даних (можливо, з помилкою) знайшли значення вузла. Рішення оберненої задачі зводиться до знаходження матриці нульових коефіцієнтів сплайнів. Зазвичай для цього використовується метод найменших квадратів (MNC).

Припустимо, що емпіричні дані представляють собою суму деякого фрактала  $\theta$  і некоректною випадкової компоненти  $\varepsilon$ :

$$Y = \theta + \varepsilon$$

Припустимо, що фрактальна  $f$ -ія з достатньою точністю  $\delta$  апроксимується фрактальним сплайном.:

$$\|K^f S - \theta\| \leq \delta$$

Рішення оберненої задачі для МНК полягає в пошуку вектора параметрів:

$$A' = (P^T P)^{-1} P^T Y,$$

де  $P$  – фрактальна матриця.

Як видно, МНК-рішення для фрактального сплайна нічим не відрізняється від

класичного випадку. Особливістю є специфічна форма матриці  $P$ . Матриця розжієннa і самоподібна, тому перевага віддається розрахунками. А саме: оскільки еквівалент фрактального сплайна з  $(R + 1)$  вузлами є звичайним сплайном з вузлами, а алгоритм розрахунку оберненої матриці має складність  $O(n^3)$ , то фрактальний сплайн дає економію приблизно в  $3k$  раз.

Виявлено залежність між якістю апроксимації і положенням сплайнів вузлів, дані рекомендації з вибору користувачем розташування вузлів материнської сплайнової лінії. У зв'язку з цим запропоновано алгоритм автоматичного пошуку оптимальної мережі вузлів.

Оптимізація фрактального сплайна. Оптимальне розташування вузлів фрактального сплайна є необхідною умовою для отримання всіх переваг самодостатності. В іншому випадку, можливі навіть ситуації, коли звичайний кубоїдний сплайн краще наближає фрактальні дані, ніж фрактальний сплайн. Проблема ускладнюється наявністю зв'язку між кількістю фрагментів, їх шириною і кількістю вбудованих шкал, що робить задачу нелінійною.

Пошук оптимальної сітки одиниць фрактального сплайна здійснюється за методом координатно-сплайна спуску, що дозволяє враховувати обмеження по близькості розташування вузлів і досягненню заданого масштабу. Алгоритм циклічно виконує оптимізацію вузлів з другого на передостанній:

$$\min \Psi(x_i, \{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_R\}), i = 2 : R - 1,$$

де  $\Psi()$  - цільова функція.

Обмеження області пошуку за сегментами забезпечує впорядкованість університетів і доступність даних по кожному сегменту. Точність позиції колишнього казначейства обмежена за своїми розмірами з метою скорочення обсягів розрахунків.

Численні експерименти з оптимізації фрактальних сплайнів вузлів дозволяють припустити, що процедура стабільна, незважаючи на зворотний зв'язок через обмеження. Оптимізація має тенденцію переміщати вузли в області з особливими властивостями (розриви і різкі зміни). Слід зазначити, що результати оптимізації для різних вихідних мереж різні, хоча і дуже близькі. Однак, це можна пояснити

впливом перешкод на деякі ідеальні мережі вузлів.

Алгоритм оптимізації для вузлів фрактального сплайна включає в себе наступні кроки:

1. Сформуйте вихідний набір вузлів і створіть посилання нульового сканування.
2. Побудувати фрактальную сплайна шкалу  $k$ .
3. Розрахувати похибку регресії для фрактального сплайна.
4. Якщо похибка регресії більше заданого значення, обчисліть нове положення сплайнів вузлів з 2-го п.2.
5. Перевірте, чи виконані умови для близького розташування вузлів для нового положення вузлів. Якщо умова не виконана, перерахуйте вузли.
6. Повторюйте дії на стр.2-5 до тих пір, поки не буде знайдена мінімальна помилка регресії.

Прикладом оптимізаційного алгоритму для апроксимації даних обмінного курсу "євродолар" за період з 04.01.1999 по 19.04.2007 р є кубічний фрактальний сплайн з 8 вузлами, як показано на рисунку.2.8. Поліпшення якості апроксимуляції помітно вже при візуальному порівнянні. Значення критерію найменшої погрішності зменшилася в деяких випадках на 40%.

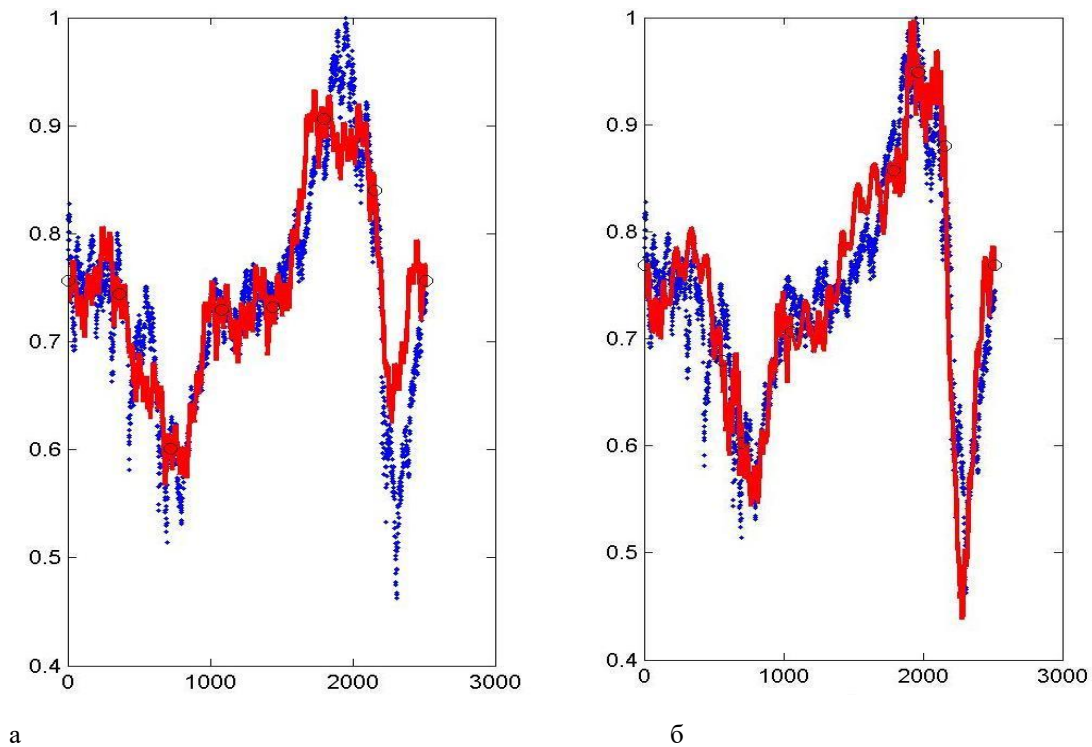


Рисунок 2.8 – Неоптимальне та оптимальне розміщення  
(сині лінії – дані валютного курсу, червоні точки – фрактальний сплайн)

Оцінка фрактального виміру. Для оцінки фрактального розміру часового ряду можна використовувати фрактальний трикутний сплайн. Даний *me-tod* є модифікацією методу побудови сегментів на основі кусково-лінійної інтерполяції. Його перевагою є простота і точність розрахунків.

Нехай фрактальна функція  $f(t)$  розмірності  $D$  описує певний часовий ряд, що містить  $2M + 1$  рівновіддалені значення. За відомими значеннями функції в точках  $i = 0, 1, 2, \dots, 2k$  будується лінійна інтерполяція фрактальної сплайнової шкали  $k$  (рисунок 2.9).

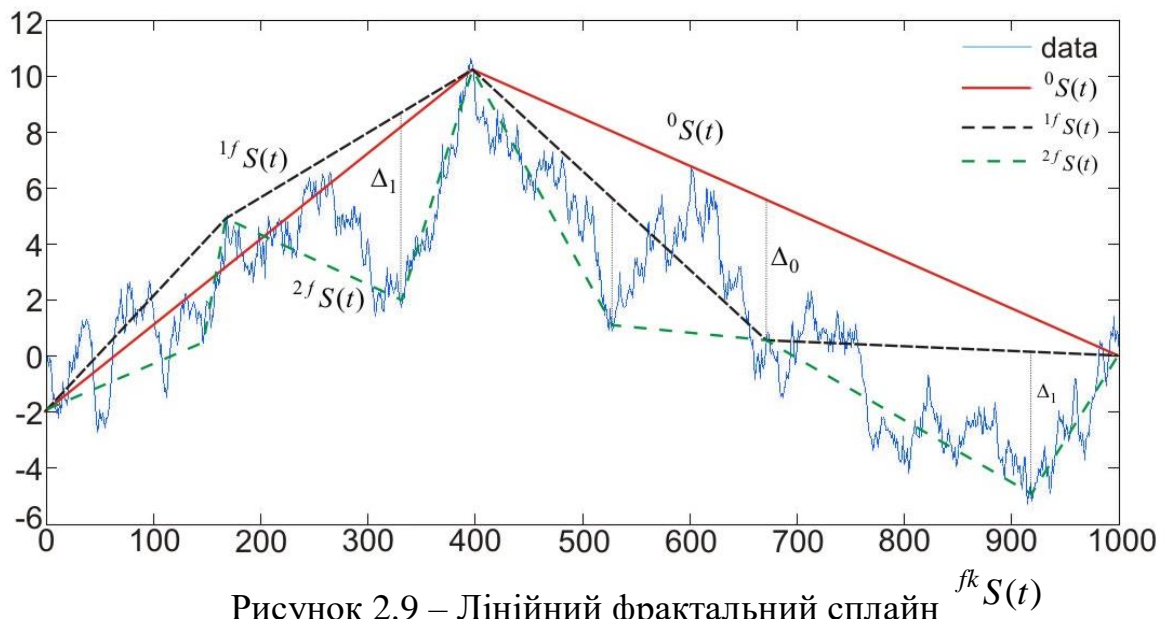


Рисунок 2.9 – Лінійний фрактальний сплайн

Помилка інтерполяції сплайна  $f^k S(t)$  відносно фрактального сплайна наступного масштабу  $f^{k+1} S(t)$  визначається як сума абсолютних відхилень  $f^k S(t)$  від  $f^{k+1} S(t)$  для всіх  $t_i = iT / 2^{k+1} = i\varepsilon_k$ :

$$\Delta_k = \sum_{i=0}^{2^{k+1}} \left| f^{k+1} S(f(t_i)) - f^k S(f(t_i)) \right| = \sum_{i=1+2}^{2^{k+1}} \left| f(t_i) - \frac{f(t_i - \varepsilon_k) + f(t_i + \varepsilon_k)}{2} \right|.$$

Значення  $\Delta_k$  пропорційні  $(\varepsilon_k)^{1-D}$  для великих  $k$ . Потім фрактальне вимір  $D$  функції  $f(t)$  можна обчислити як нахил  $s$ -лінії, яка описує залежність  $\Delta_k$  від  $k$  у логарифмічному масштабі, тобто

$$D = 1 + s / \log 2$$

Метод верифікується на фрактальних функціях з відомими розмірами. Точні значення розмірів отримані з даними  $N = 5000$ , для меншої кількості даних ( $N \geq 1500$ ) є середня похибка 4%.

## 4. Засоби реалізації програмного забезпечення

Програмна реалізація системи виконана згідно сучасних вимог до розробки програмного забезпечення . Перш за все було обрано засоби розробки, а саме мову програмування, також було спроектовано діаграму класів майбутньої програмної системи. Весь продукт було поділено на наступні програмні модулі: модуль побудови графіку будь-якої компанії, модуль прогнозування на основі теорії фракталів.

### ***4.1 Вибір середовища створення програмного продукту***

JavaScript часто скорочується як JS, це високорівнева, інтерпретується мова програмування, що відповідає специфікації ECMAScript. JavaScript має синтаксис фігурних дужок, динамічну типізацію, об'єктно-орієнтаційну орієнтацію на основі прототипу та функції першого класу.

Поряд з HTML і CSS, JavaScript є однією з основних технологій World Wide Web. JavaScript дозволяє інтерактивні веб-сторінки і є важливою частиною веб-додатків. Переважна більшість веб-сайтів використовують її, а головні веб-браузери мають спеціальний механізм JavaScript для його виконання.

Як багатопартійний мова, JavaScript підтримує функціональні та імперативні (включаючи об'єктно-орієнтовані та прототипні) стилі програмування. Вона має API для роботи з текстом, масивами, датами, регулярними виразами та DOM, але сама мова не включає жодного вводу-виводу, наприклад мереж, сховищ або графічних засобів. Він покладається на хост-середовище, в яке він вбудований, щоб забезпечити ці функції.

Спочатку реалізовані клієнтські веб-браузери, двигуни JavaScript тепер вбудовані в багато інших типів програмного забезпечення, включаючи серверні веб-сервери та бази даних, а також у не-веб-програми, такі як текстові процесори та програмне забезпечення PDF, а також під час виконання середовища, які роблять

JavaScript доступним для написання мобільних і настільних додатків, включаючи віджети робочого столу.

Терміни Vanilla JavaScript і Vanilla JS посилаються на JavaScript, не розширений жодними рамками або додатковими бібліотеками. Сценарії, написані на Vanilla JS, є звичайним JavaScript кодом.

Незважаючи на наявність подібності між JavaScript і Java, включаючи назву мови, синтаксис і відповідні стандартні бібліотеки, ці дві мови відрізняються і відрізняються значним дизайном. На JavaScript впливали такі мови програмування, як Self і Scheme. Формат серіалізації JSON, який використовується для зберігання структур даних у файлах або передачі їх по мережах, заснований на JavaScript.

Функція першого класу; функція вважається об'єктом. Таким чином, функція може мати властивості і методи, такі як `.call ()` і `.bind ()`. Вкладена функція - це функція, визначена в межах іншої функції. Він створюється кожного разу, коли викликається зовнішня функція. Крім того, кожна вкладена функція утворює лексичне закриття: лексична область зовнішньої функції (включаючи будь-яку константу, локальну змінну або значення аргументу) стає частиною внутрішнього стану кожного внутрішнього об'єкта функції, навіть після виконання зовнішньої функції. JavaScript також підтримує анонімні функції.

Технології сценаріїв Microsoft, включаючи VBScript і JScript, були випущені в 1996 році. JScript, зворотньо-інженерна реалізація JavaScript Netscape, була частиною Internet Explorer 3. JScript також був доступний для сценаріїв на сервері в Internet Information Server. Internet Explorer 3 також включив першу підтримку Microsoft для CSS і різних розширень до HTML, але в кожному випадку реалізація була помітно іншою, ніж у Netscape Navigator. Ці відмінності ускладнювали дизайнерів і програмістів для того, щоб один веб-сайт працював добре в обох браузерах, що призвело до використання "найкращого перегляду в Netscape" і логотипів "кращого перегляду в Internet Explorer", які характеризували ці перші роки війн у браузері. JavaScript почав набувати репутацію однієї з перешкод для крос-платформного та керованого стандартами Web. Деякі розробники взяли на себе важке завдання, намагаючись зробити свої сайти працювати в обох основних браузерах, але багато хто



не міг дозволити собі час. З виходом Internet Explorer 4, Microsoft представила концепцію Dynamic HTML, але відмінності в реалізації мови і різних і власних об'єктних моделях документів залишилися і були перешкодами для широкого використання JavaScript в Інтернеті.

## **4.2    *Архітектура програмного забезпечення***

При розробці використовувалась мова програмування JavaScript та фреймворк Angular, яка надає готові архітектурні рішення. Angular - це платформа і основа для побудови клієнтських додатків на TypeScript. Angular написаний на TypeScript. Вона реалізує основні та додаткові функції як набір бібліотек TypeScript, які ви імпортуєте у свої програми.

Основними будівельними блоками програми Angular є NgModules, які забезпечують контекст компіляції для компонентів. NgModules збирають відповідний код у функціональні набори; Угловое додаток визначається набором NgModules. Додаток завжди має принаймні кореневий модуль, який дозволяє завантажувати початкову програму, і зазвичай має багато інших модулів.

Компоненти визначають види, які є набором елементів екрану, які Angular може вибрати і змінити відповідно до логіки програми та даних.

Компоненти використовують послуги, які забезпечують певну функціональність, безпосередньо не пов'язану з переглядами. Постачальники послуг можуть бути введені в компоненти як залежності, що робить ваш код модульним, багаторазовим і ефективним.

Як компоненти, так і служби - це просто класи, де декоратори позначають їх тип і надають метадані, які повідомляють Angular, як їх використовувати.

Метадані для класу компонентів асоціюють його з шаблоном, який визначає подання. Шаблон поєднує звичайний HTML з директивами Angular та розміткою, що дозволяє Angular змінити HTML перед його відображенням.

Метадані для класу послуг надають інформацію Angular, яка повинна зробити її доступною для компонентів через ін'єкцію залежностей (DI).

Компоненти програми зазвичай визначають багато переглядів, організованих ієрархічно. Angular надає послугу маршрутизатора, щоб допомогти вам визначити шляхи навігації серед переглядів. Маршрутизатор забезпечує складні навігаційні можливості браузера.

Кутові модулі NgModules відрізняються від і доповнюють модулі JavaScript (ES2015). NgModule оголошує контекст компіляції для набору компонентів, призначених домену домену, робочому процесу або тісно пов'язаному набору можливостей. NgModule може асоціювати свої компоненти з відповідним кодом, таким як послуги, для формування функціональних одиниць.

Кожне додаток Angular має кореневий модуль, який умовно називається AppModule, який забезпечує механізм завантаження, який запускає програму. Додаток зазвичай містить багато функціональних модулів.

Подібно модулям JavaScript, NgModules можуть імпортувати функціональні можливості інших NgModules і дозволяють експортувати та використовувати їхні власні функціональні можливості іншими NgModules. Наприклад, щоб використовувати службу маршрутизатора у вашому додатку, ви імпортуєте маршрутизатор NgModule.

Упорядкування коду в окремі функціональні модулі допомагає в управлінні розробкою складних додатків, а також у розробці для повторного використання. Крім того, ця методика дозволяє скористатися перевагою лінивого завантаження, тобто завантаження модулів на вимогу, щоб мінімізувати кількість коду, який необхідно завантажувати під час запуску.доступ.

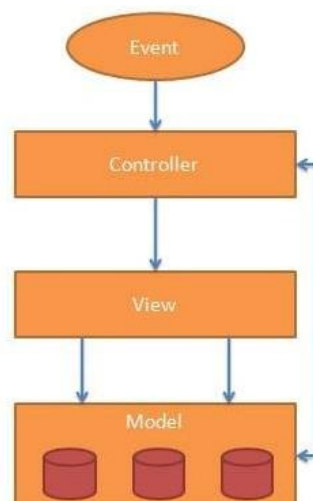


Рисунок 4.1 – Приклад архітектури Angular

### 4.3 Діаграма класів

Діаграма класів програмної системи моделювання ізотропних кривих і криволінійних обводів, плоских сіток й ізометричних поверхонь на основі кривих Без'є третього порядку представлена на рисунку 4.2. Діаграма класів повністю відтворює структуру розробленого програмного забезпечення.

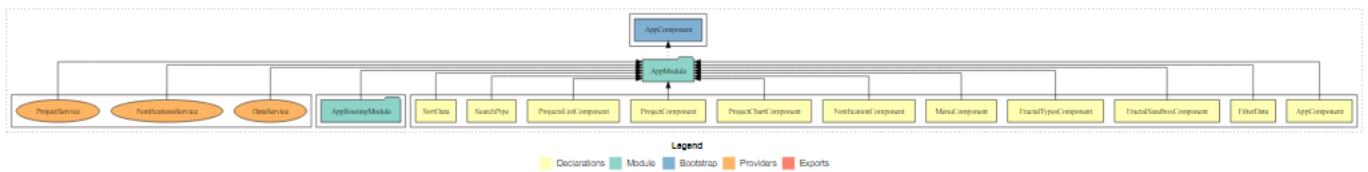


Рисунок 4.2 – Діаграма класів

#### **4.4 Структура БД**

В якості роботи з базою даних, обрали MongoDB. Це програмне забезпечення для роботи з документами. Класифікований як програма для бази даних NoSQL, MongoDB використовує JSON-подібні документи зі схемою. MongoDB, розроблений MongoDB Inc. і ліцензований під ліцензією на відкритий сервер (SSPL).списку. Існують спеціальні запити.

MongoDB підтримує поля, запит діапазону і пошук регулярних виразів. Запити можуть повертати конкретні поля документів, а також включати визначені користувачем функції JavaScript. Запити також можуть бути налаштовані для повернення випадкової вибірки результатів заданого розміру.

Поля в документі MongoDB можуть бути індексовані за допомогою первинних і вторинних індексів.

Mongo забезпечує високу доступність з наборами реплік. Набір реплік складається з двох або більше копій даних. Кожен член набору реплік може діяти в ролі первинної або вторинної репліки в будь-який час. Всі записи і читання виконуються за замовчуванням на первинній репліці. Вторинні репліки підтримують копію даних первинного з використанням вбудованої реплікації. Коли первинної репліки не вдається, набір реплік автоматично проводить виборчий процес, щоб визначити, який вторинний повинен стати основним. Вторинні можуть додатково служити операції читання, але ці дані в кінцевому рахунку узгоджуються лише за замовчуванням.

MongoDB масштабується горизонтально, використовуючи шардінг. Користувач вибирає ключ shard, який визначає, як будуть поширюватися дані в колекції. Дані розбиваються на діапазони (на основі клавіші shard) і розподіляються по декільком осколкам. (Шард - майстер з однією або кількома репліками.). Альтернативно, клавіша Shard може бути хеширована для відображення на шарді, що забезпечує рівномірний розподіл даних.

Може працювати на декількох серверах, балансуючи навантаження або дублюючи дані, щоб підтримувати роботу системи в разі збою в роботі обладнання.

## Зберігання файлів

Вона може використовуватися як файлова система, звана GridFS, з функціями балансування навантаження та реплікації даних на декількох машинах для зберігання файлів.

Ця функція, яка називається файловою системою grid, входить до драйверів MongoDB. MongoDB надає розробникам функції для маніпулювання файлами та вмісту. Доступ до GridFS можна отримати за допомогою утиліти mongofiles або плагінів для Nginx і lighttpd. GridFS ділить файл на частини або шматки, і зберігає кожен з цих фрагментів як окремий документ.

MongoDB надає три способи виконання агрегації: консолідаційний конвеєр, функцію зменшення карти і одноцільові агрегаційні методи.

Map-reduction може використовуватися для пакетної обробки даних і операцій агрегації. Але згідно з документацією MongoDB, трубопровід агрегації забезпечує кращу продуктивність для більшості операцій агрегації.

Структура агрегації дозволяє користувачам отримувати типи результатів, для яких використовується пропозиція SQL GROUP BY. Оператори агрегації можуть бути зв'язані між собою, щоб сформувавши трубопровід - аналогічний трубам Unix. Система агрегації включає оператора \$ lookup, який може об'єднувати документи з декількох документів, а також статистичні оператори, такі як стандартне відхилення.

JavaScript може використовуватися в запитах, функціях агрегації (наприклад, MapReduce) і надсилатися безпосередньо до бази даних для виконання.

Так як база даних програмного забезпечення складає дві таблиці(рисунок 4.3-4.4), то дуже вдала ідея використовувати саме MongoDB.

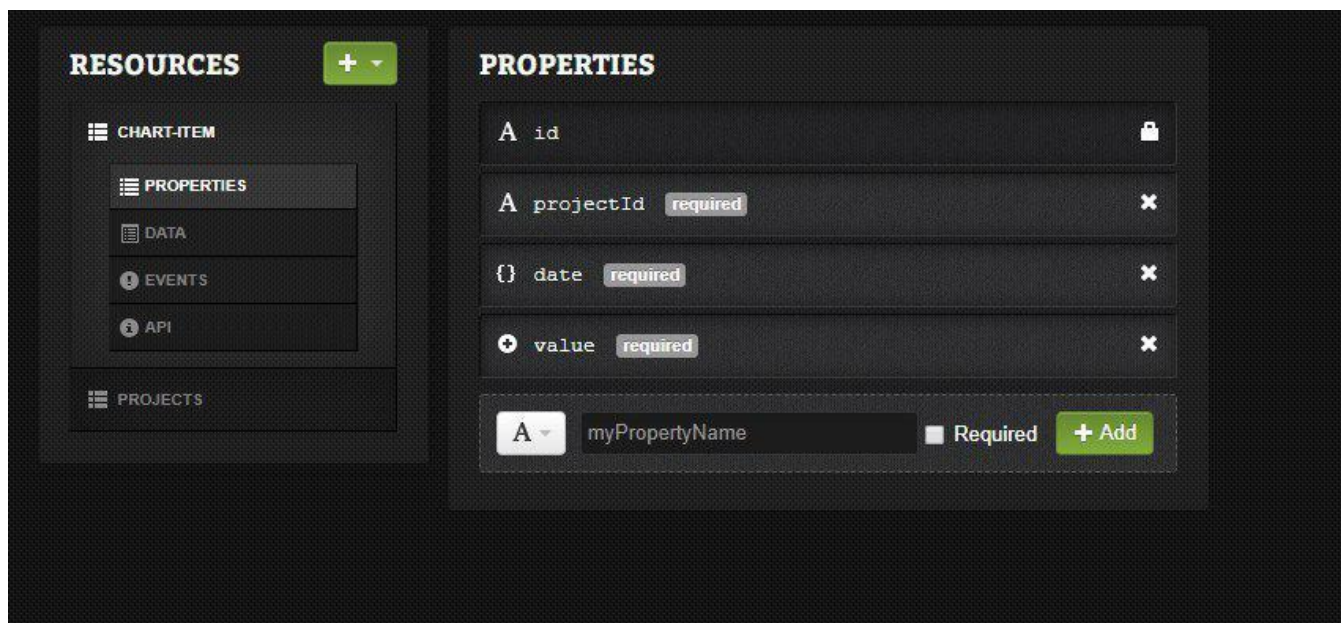


Рисунок 4.3 – Перша таблиця в БД

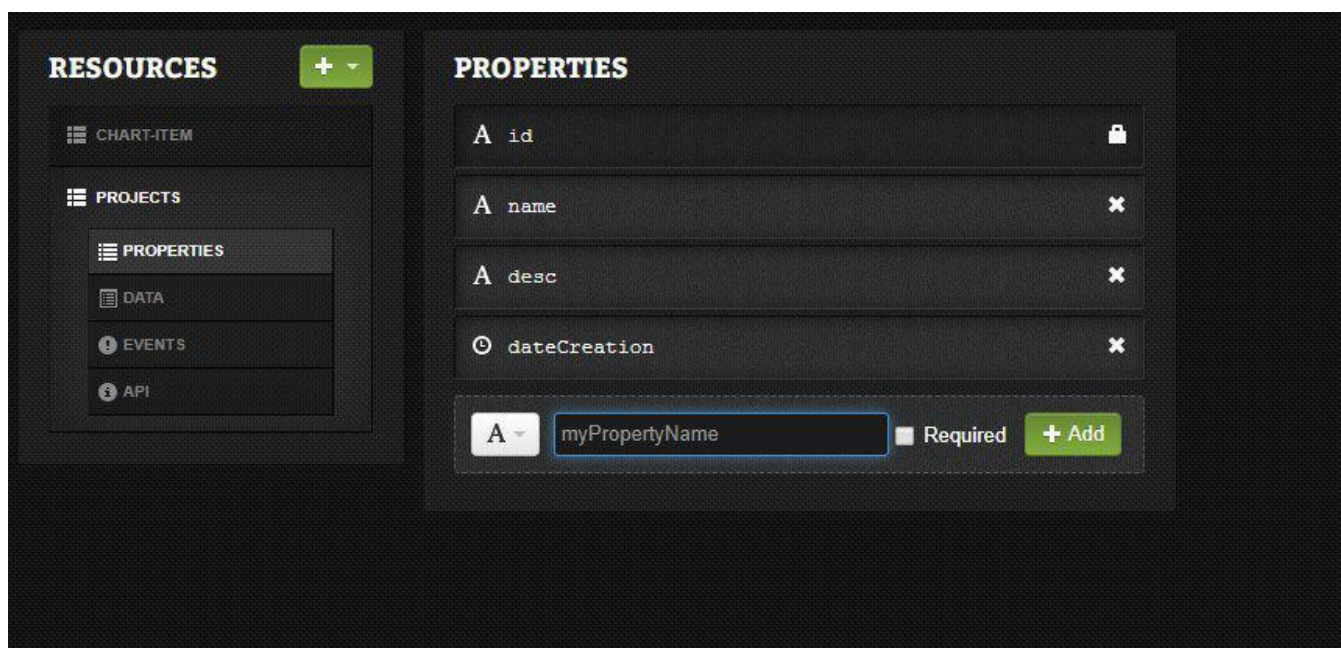


Рисунок 4.4 – Друга таблиця в БД

## 5. Інтерфейс роботи з користувачем

При розробці програмного забезпечення для кінцевого користувача, головний акцент було зроблено на простоті та зручності під час користування програмним забезпеченням.

### 5.1 Системні вимоги

Для коректної роботи розробленої програми на персональному комп'ютері має бути встановлений Node.js, npm, Mongo.db, Angular-cli і звичайно будь-який браузер.

### 5.2 Робота користувача з програмною системою

При завантаженні програмного додатку з'являється головне вікно (рисунок 5.1), на якому розташовані компанії (рисунок 5.1). Зверху знаходиться панель вкладок, що надає можливість обрати, з чим саме користувач хоче працювати: додавання нової компанії та редагування старих даних чи видалення компанії з системи, або вивести на екран види фракталів, або побудувати фрактал Серпинського з різною кількістю трикутників. При побудові графіка можна вибрати як буде відображатися, або в місяцях, або ж в роках.

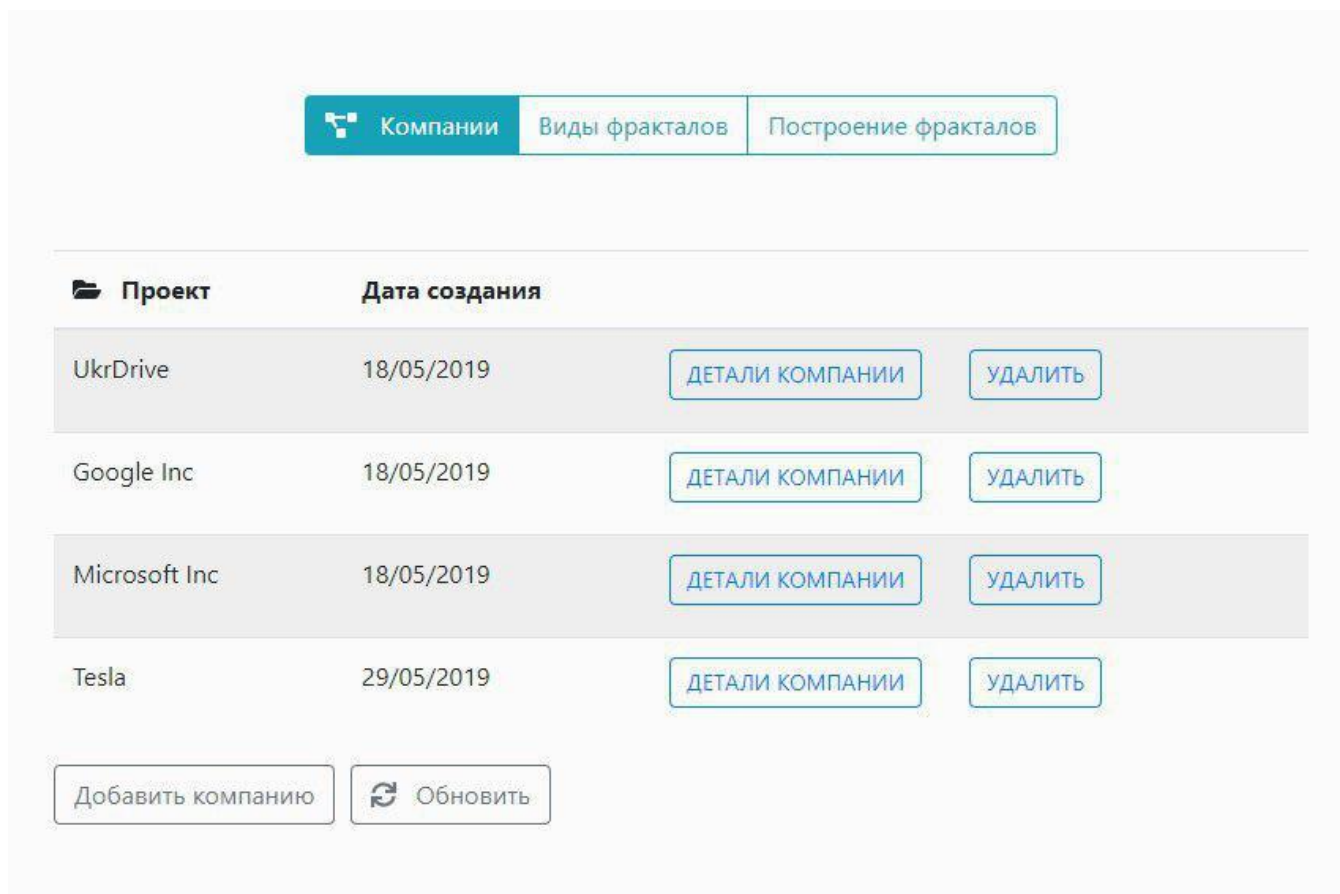


Рисунок 5.1 — Головне вікно програми

Головне вікно програми(рисунок 5.1) містить список компаній, час створення її, також можна одразу дізнатися про компанію певну інформацію та якщо необхідно вилучити її з бази разом з усіма даними.



Компании

Виды фракталов

Построение фракталов

## Добавить новую компанию

Название компании

ОПИСАНИЕ:

Описание отсутствует

Отменить

Сохранить

Рисунок 5.2 — Додавання нового графіку(нової компанії)

Також можна додавати нову компанію(рисунок 5.2), та за необхідністю писати кратку інформацію про неї.

Компании

Виды фракталов

Построение фракталов

Microsoft Inc

Данные графика:

20/6/2019 : 5

Добавить новый элемент графика:

Значение (\$)

Дата:

yyyy-mm-dd

Добавить

ОПИСАНИЕ:

Описание отсутствует

Отменить

Сохранить

Рисунок 5.3 — Додавання певних значень для аналізу та побудови графіку

Зверху на сторінці розташовані дані графіка(рисунок 5.3), які були додані раніше , що слугують для побудови графіка по значенням. Нижче треба вказувати конкретні значення та дату коли були ці значення. А далі йде побудова графіка на основі цих значень(рисунок 5.4).

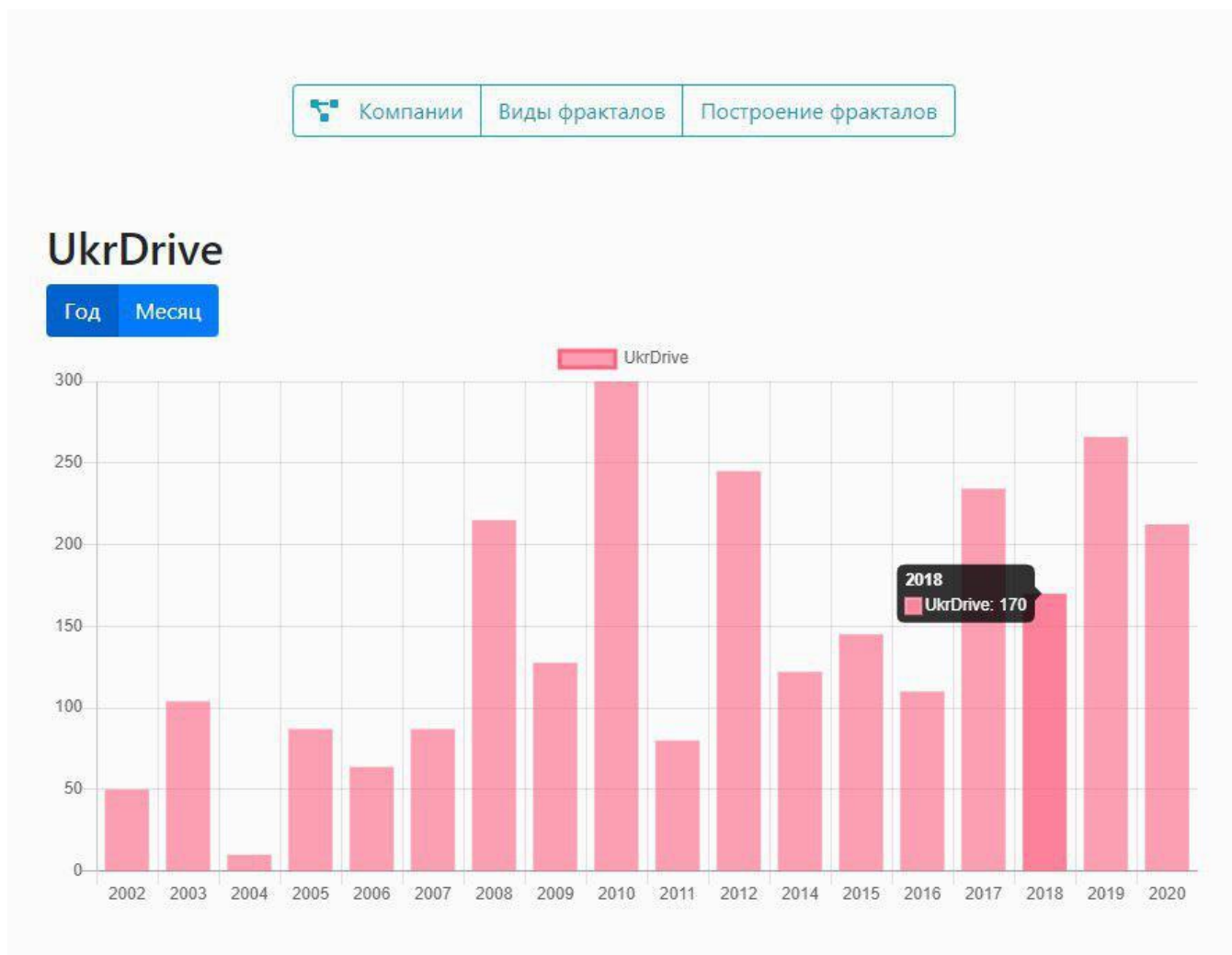
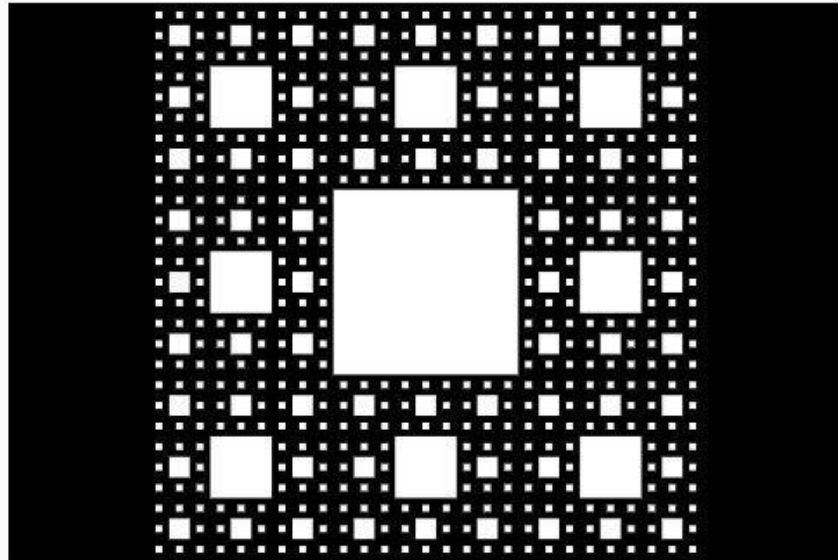


Рисунок 5.4 — Пример аналізу графіка з певними значеннями.

Також існують дві кнопки в горі(рисунок 5.4), що дозволяють зробити аналіз не тільки по рокам, але й по місяцям. Далі переходимо до видів фракталів, що теж важливо.(рисунок 5.5-5.7)

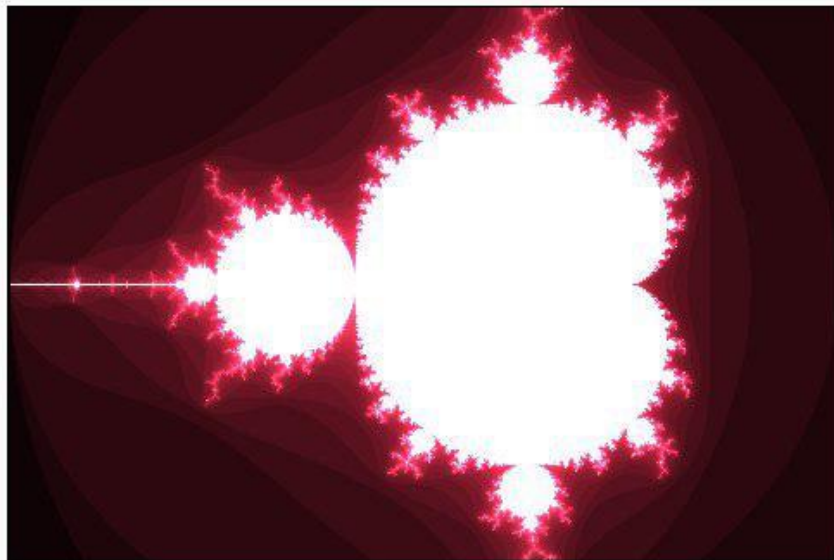
[Компании](#)[Виды фракталов](#)[Построение фракталов](#)

Ковер Серпинского (Геометрический фрактал)

[Геометрический](#)[Алгебраический](#)[Стохастический](#)

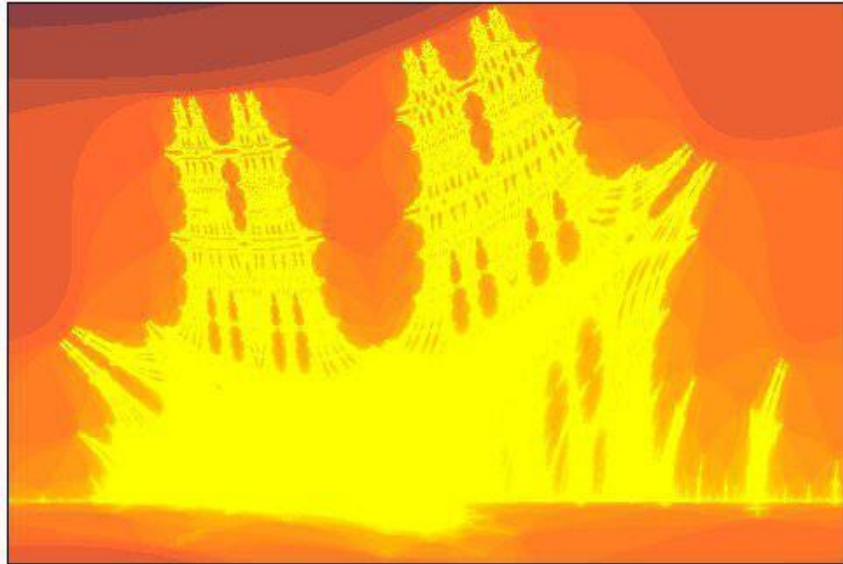
Рисунок 5.5 — Пример геометрического фрактала

На сторінці види фракталів представлено 3 різних фрактали (геометричний, стохастичний та алгебраїчний). За допомогою кнопок ми можемо переключати їх.



Фрактал Мандельбронта (Алгебраический фрактал)

Рисунок 5.6 — Пример алгебраического фрактала



Горящий корабль (Стохастический фрактал)

Геометрический

Алгебраический

Стохастический

Рисунок 5.7 — Пример стохастического фрактала

Також можна створювати самому фрактали з певною кількістю трикутників(рисунок 5.8)

Вложенность:



5

Количество треугольников: 243

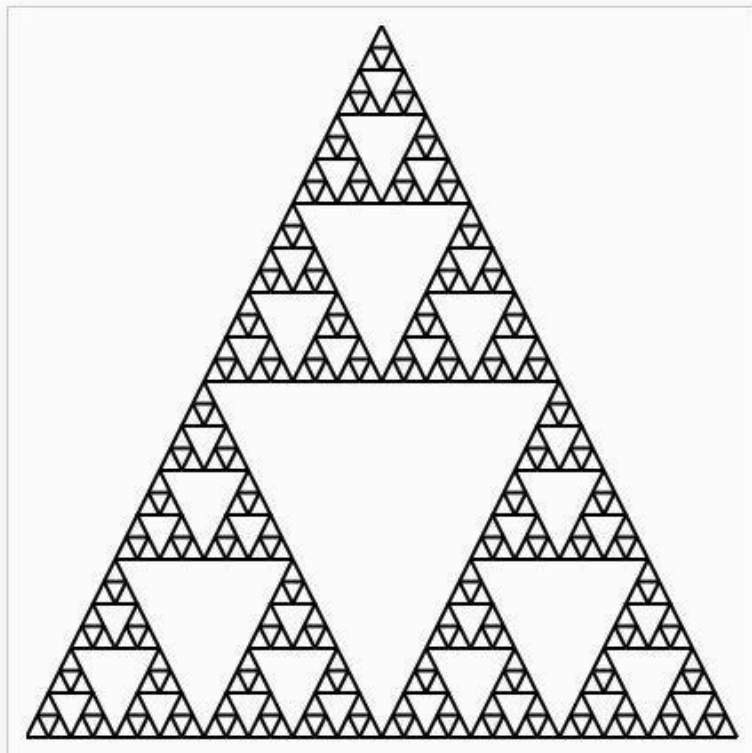


Рисунок 5.8 — Побудова трикутників Серпинського

# ВИСНОВКИ

Роботу присвячено актуальному напрямку прогнозуванню економічних процесів на основі теорії фракталів. При вирішенні поставлених задач були отримані наступні результати:

1. Проаналізовано сучасні методи для аналізу та прогнозування економічних процесів та багатомасштабних часових рядів з застосуванням фракталів і сплайнів.
2. Проаналізовано програмне забезпечення для розв'язання задач прогнозування.
3. Обрано засоби реалізації для прогнозування соціально-економічних процесів і обрано алгоритм побудови різних видів фрактальних сплайнів.
4. Розроблено структуру та архітектуру програмного забезпечення.
5. Розроблено концептуальну модель бази даних.
6. Розроблено програмне забезпечення та проведено моделювання розроблених алгоритмів аналізу.

Програмну систему розроблено мовою програмування JavaScript та фреймворку Angular.

Створений проект виконує прогнозування соціально-економічних процесів на основі фрактального аналізу, модулює графік та тренд будь-якої компанії .



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Shelevytsky I. Fractal Spline Signals: Synthesis and Estimation / I.Shelevytsky, O.Novikova // IEEE XXXIII International Scientific Conference “Electronics and Nanotechnology (ELNANO)”, Kiev, April 16-19, 2018. – P. 469-472.
2. Новікова О.Б. Використання фрактальних сплайнів для обробки сигналів складної форми / О.Б.Новікова // Матеріали 40-ї міжнародної наукової конференції «Питання оптимізації обчислень (ПОО-XL)», Кацівелі, 30 вересня – 4 жовтня 2013 р. – Кацівелі, 2013. – С.186-187.
3. Novikova O. Modeling of multiscale time series using fractal splines / O.Novikova, V.Shutko//International Conference CSE-2013, Lviv, November 21-23, 2013. –P.118-119.
4. Новікова О.Б. Нерегулярні фрактальні сплайни та їх застосування до обробки сигналів та часових рядів / О.Б.Новікова // Матеріали IV міжнародної науково-практичної конференції «Обробка сигналів і негаусівських процесів», Черкаси, 22-24 травня 2013 р. – Черкаси, 2013. – С.103-104.
5. Новікова О.Б. Фрактальні сплайни у прогнозуванні валютних курсів / О.Б.Новікова // Матеріали I міжнародної науково-практичної конференції «Сучасні соціально-економічні системи: тенденції розвитку», Кривий Ріг, 12-14 травня 2011 р. – Кривий Ріг, 2011. – С. 88-90.

6. Новікова О.Б. Дослідження часових рядів валютних курсів за допомогою фрактальних сплайнів / О.Б.Новікова // Матеріали V Міжнародної конференції молодих вчених CSE-2011 «Комп'ютерні науки та інженерія», Львів, 24-26 листопада 2011р. – Львів, 2011. – С.182-183.
7. Новікова О.Б. Розробка програмного забезпечення для фрактального аналізу / О.Б.Новікова // Матеріали V міжнародної науково-практичної конференції «Методи, моделі та інформаційні технології в управлінні соціально-економічними, екологічними та технічними системами», Луганськ-Євпаторія, 17-19 жовтня 2012 р. – Луганськ-Євпаторія, 2012. – С.143-145.
8. Новікова О.Б. Новий підхід до побудови фрактальних сплайнів / О.Б.Новікова // Матеріали III міжнародної науково-практичної конференції «Системний аналіз. Інформатика. Управління», Запоріжжя, 14-16 березня 2012 р. – Запоріжжя, 2012. – С.32-33.

## **Додаток А**

## Прогнозування на основі теорії фракталів

### Специфікація

УКР.НТУУ”КПІ ім. Ігоря Сікорського”\_ТЕФ\_АПЕПС\_ТМ52

Аркушів 2

Київ 2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТМ52_1 9Б 81-1	Записка.docx	Пояснювальна записка

Компоненти		
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТМ52_1 9Б 12-1	app.module.ts	Головний модуль додатку
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТМ52_1 9Б 13-2	Додаток В.doc	Опис програмного модуля

## **Додаток Б**

Прогнозування на основі теорії фракталів.

Текст програми

УКР.НТУУ"КПІ ім. Ігоря Сікорського" \_ТЕФ\_АПЕПС\_ТМ52\_19Б 13-1

Аркушів 8

Київ 2019

```
//Список компаній
<div *ngIf="errorMessage" class="alert alert-danger">
  {{errorMessage}}
</div>
<table *ngIf="!!projects.length" class="table table-striped" @appearanceAnimation>
  <thead>
    <tr>
      <th><i class="fas fa-folder-open"></i> Проект</th>
      <th></th>
      <th>Дата создания</th>
      <th></th>
    </tr>
  </thead>
  <tr *ngFor="let project of projects">
    <td class="project-list-td"
      [routerLink]="['/chart', project.id]'
      [queryParams]='{'project': project.name}'>{{project.name}}</td>
    <td>
      <p>{{project.dateCreation | date: 'dd/MM/yyyy'}}</p>
    </td>
    <td>
      <button [routerLink]="['/projects', project.id]" class="btn btn-outline-primary">Детали
компаний</button>
      <button (click)="deleteProject(project)" class="btn btn-outline-primary">Удалить</button>
    </td>
  </tr>
</table>

<div>
  <button class="btn btn-outline-secondary add-company" routerLink="/new-project">Добавить
компанию</button>
  <button class="btn btn-outline-secondary" (click)="getProjects()"><i class="fas fa-sync"></i>
Обновить</button>
</div>
```

```
//Побудова графіку
import {Component, OnDestroy, OnInit} from "@angular/core";
import {ActivatedRoute} from '@angular/router';
import {ProjectService} from '../project/project.service';
import {ChartDataEnum} from '../enums/chart-data.enum';
import {ChartItemInterface} from '../interfaces/chart-item.interface';
import {MyFractalPredictionService} from '../services/my-fractal-prediction.service';
import {FormBuilder, FormGroup} from '@angular/forms';
import {untilDestroyed} from "ngx-take-until-destroy";
```

```
@Component({
  selector: 'app-project-chart',
  templateUrl: './project-chart.component.html',
  styleUrls: ['./project-chart.component.css']
})
export class ProjectChartComponent implements OnInit, OnDestroy {
  public radioGroupForm: FormGroup;

  readonly minItems = 12;
  constructor(
    private route: ActivatedRoute,
    private formBuilder: FormBuilder,
    private projectService: ProjectService,
```

```

    private predictService: MyFractalPredictionService,
  ) {
  }

  title: string;
  chartDataType = ChartDataEnum.Year;

  public barChartOptions = {
    scaleShowVerticalLines: false,
    responsive: true,
    scales: 'time'
  };
  public barChartLabels: [string] = [];
  public barChartType = 'bar';
  public barChartLegend = true;
  public barChartData = [{ data: [], label: ''}];
  projectChartItems = [];

  ngOnInit() {
    this.radioGroupForm = this.formBuilder.group({
      'viewMode': 0
    });

    // Смена вида отображения графика
    this.radioGroupForm.controls.viewMode.valueChanges
      .pipe(untilDestroyed(this))
      .subscribe(
        res => {
          this.chartDataType = res;
          const predictedValues = this.predictService.getPredictValue(this.projectChartItems);
          this.updateChart([...this.projectChartItems, ...predictedValues]);
        }
      );

    // Получение данных графика
    this.route.params
      .pipe(untilDestroyed(this))
      .subscribe(routeParams => {
        this.projectService.getProjectChartItems().pipe(
          untilDestroyed(this)
        ).subscribe(
          items => {
            this.projectChartItems = items.filter(item => item.projectId === routeParams.id);
            // Получение прогноза на 12 месяцев
            if (this.projectChartItems.length >= this.minItems) {
              const predictedValues = this.predictService.getPredictValue(this.projectChartItems,
12);
              this.updateChart([...this.projectChartItems, ...predictedValues]);
            }
          },
        );
      });

    this.route.queryParams.pipe(
      untilDestroyed(this)
    ).subscribe(res => {
      this.title = res.project;
    });
  }

  ngOnDestroy() {
  }

  // Обновление графика
  updateChart(items: ChartItemInterface[]): void {

```

```

const uniqueYears = new Set<number>();
const uniqueValues = [];

// Если выбран режим по годам
if (this.chartDataType === ChartDataEnum.Year) {
  items.forEach(
    item => {
      uniqueYears.add(item.date.year);
    }
  );

  uniqueYears.forEach(
    year => {
      uniqueValues.push(this.findValue(year, items));
    }
  );

  this.barChartLabels = Array.from(uniqueYears);

  this.barChartData = [{
    data: uniqueValues,
    label: this.title,
  }];

// Если выбран режим по месяцам
} else {
  const itemsByMonths = new Map<number, ChartItemInterface>();

  items.forEach(
    item => {
      const token = item.date.year + '/' + item.date.month;

      const savedItem = itemsByMonths.get(token);
      if (savedItem) {
        itemsByMonths.set(token, {
          ...savedItem,
          value: (savedItem.value + item.value) / 2
        });
      } else {
        itemsByMonths.set(token, item);
      }
    }
  );
  this.barChartLabels = Array.from(itemsByMonths.keys());

  this.barChartData = [{
    data: Array.from(itemsByMonths.values()).map(res => res.value),
    label: this.title,
  }];
}

findValue(year, items: ChartItemInterface[]): number {
  const yearItems = items.filter(item => +item.date.year === +year);
  let sum = 0;

  yearItems.forEach((value) => sum = sum + +value.value);

  return sum / yearItems.length;
}

//Види фракталів
import {Component, OnInit} from "@angular/core";

```



```

@Component({
  selector: "app-fractal-types",
  templateUrl: "../fractal-types.component.html",
  styleUrls: ["../fractal-types.component.css"]
})
export class FractalTypesComponent implements OnInit {

  constructor() {
  }

  ngOnInit() {
    let canvas,
        context,
        max_iterations;
    const textEl = document.getElementById("frac-name");

    function checkCanvasIsSupported() {
      canvas = document.getElementById("canvas");
      canvas.width = 480;
      canvas.height = 320;
      context = canvas.getContext("2d");
      render();

      document.getElementById("geometric").addEventListener("click", () => {
        clearRect();
        // draw Sierpinski carpet
        drawSierpinskiCarpet();
        textEl.innerText = "Ковер Серпинского (Геометрический фрактал)";
      });

      document.getElementById("algebraic").addEventListener("click", () => {
        clearRect();
        // draw Mandelbrot fractal
        drawMandelbrot();
        textEl.innerText = "Фрактал Мандельбронта (Алгебраический фрактал)";
      });

      document.getElementById("stochastic").addEventListener("click", () => {
        clearRect();
        // draw Burning ship
        drawBurningShipFractal();
        textEl.innerText = "Горящий корабль (Стохастический фрактал)";
      });
    }

    function render() {
      clearRect();
      // draw Sierpinski carpet
      drawSierpinskiCarpet();
    }

    function clearRect() {
      context.clearRect(0, 0, canvas.width, canvas.height);
    }

    function drawMandelbrot() {
      // prepare image and pixels
      const image_data = context.createImageData(canvas.width, canvas.height);
      const d = image_data.data;

      max_iterations = 100;
      for (let i = 0; i < canvas.height; i++) {
        for (let j = 0; j < canvas.width; j++) {

          // limit the axis

```

```

    const x0 = -2.0 + j * 3.0 / canvas.width;    // (-2, 1)
    const y0 = -1.0 + i * 2.0 / canvas.height;  // (-1, 1)

    let x = 0;
    let y = 0;
    let iteration = 0;

    while ((x * x + y * y < 4) && (iteration < max_iterations)) {
        const x_n = x * x - y * y + x0;
        const y_n = 2 * x * y + y0;
        x = x_n;
        y = y_n;
        iteration++;
    }

    // set pixel color [r,g,b,a]
    d[i * canvas.width * 4 + j * 4 + 0] = iteration * 15;
    d[i * canvas.width * 4 + j * 4 + 1] = iteration * 3;
    d[i * canvas.width * 4 + j * 4 + 2] = iteration * 5;
    d[i * canvas.width * 4 + j * 4 + 3] = 255;
}
}

// draw image
context.putImageData(image_data, 0, 0);
}

function drawBurningShipFractal() {
    // prepare image and pixels
    const image_data = context.createImageData(canvas.width, canvas.height);
    const d = image_data.data;

    max_iterations = 100;
    for (let i = 0; i < canvas.height; i++) {
        for (let j = 0; j < canvas.width; j++) {

            const x0 = -1.80 + j * (-1.7 + 1.80) / canvas.width;
            const y0 = -0.08 + i * (0.01 + 0.08) / canvas.height;
            let x = 0;
            let y = 0;
            let iteration = 0;

            while ((x * x + y * y < 4) && (iteration < max_iterations)) {
                const x_n = x * x - y * y + x0;
                const y_n = 2 * Math.abs(x * y) + y0;
                x = x_n;
                y = y_n;
                iteration++;
            }

            // set pixel color [r,g,b,a]
            d[i * canvas.width * 4 + j * 4 + 0] = 25 + iteration * 30;
            d[i * canvas.width * 4 + j * 4 + 1] = 25 + iteration * 10;
            d[i * canvas.width * 4 + j * 4 + 2] = 85 - iteration * 5;
            d[i * canvas.width * 4 + j * 4 + 3] = 255;
        }
    }

    // draw image
    context.putImageData(image_data, 0, 0);
}

function drawSierpinskiCarpet() {
    // draw carpet
    const draw_carpet = function (x, y, width, height, iteration) {

```

```

    if (iteration === 0) return;
    const w = width / 3;
    const h = height / 3;

    // draw subsquare
    context.fillStyle = "rgb(255, 255, 255)";
    context.fillRect(x + w, y + h, w, h);

    // draw subcarpets
    for (let i = 0; i < 3; i++) {
      for (let j = 0; j < 3; j++) {
        // remove central subsquare
        if (j === 1 && i === 1) continue;
        draw_carpet(x + j * w, y + i * h, w, h, iteration - 1);
      }
    }
  };

  // init carpet size
  const carpet_width = canvas.height;
  const carpet_height = canvas.height;
  // align to the center
  const carpet_left = (canvas.width - carpet_width) / 2;
  // limit the depth of recursion
  const max_iterations = 4;

  // draw Sierpinski carpet
  draw_carpet(carpet_left, 0, carpet_width, carpet_height, max_iterations);
}

checkCanvasIsSupported();
}

```

```

//Побудова фрактала Серпинського
import { Component, OnInit } from '@angular/core';
import { FormControl } from "@angular/forms";

@Component({
  selector: 'app-fractal-sandbox',
  templateUrl: './fractal-sandbox.component.html',
  styleUrls: ['./fractal-sandbox.component.css']
})
export class FractalSandboxComponent implements OnInit {
  ctx;

  delay1;
  delay2;

  levelForm = new FormControl(0);

  constructor() { }

  ngOnInit() {
    // @ts-ignore
    this.ctx = document.getElementById("myCanvas").getContext("2d");

    this.drawFract({ x: 0, y: 400 }, { x: 200, y: 0 }, { x: 400, y: 400 }, 0);
  }

  /*Dibujar línea*/
  drawLine(p0, p1, color = "black") {
    this.ctx.beginPath();
    this.ctx.moveTo(p0.x, p0.y);

```

```

    this.ctx.lineTo(p1.x, p1.y);
    this.ctx.strokeStyle = color;
    this.ctx.lineWidth = 2;
    this.ctx.stroke();
}

/*Dibujar triángulo*/
drawTriangle(p0, p1, p2) {
    this.drawLine(p0, p1);
    this.drawLine(p1, p2);
    this.drawLine(p2, p0);
}

/*Dibujar fractal*/
drawFract = (p0, p1, p2, nivel) => {
    if (nivel > 0) {
        /*Calcular nuevos vértices intermedios*/
        const pA = {
            x: p0.x + (p1.x - p0.x) / 2,
            y: p0.y + (p1.y - p0.y) / 2
        },
        pB = {
            x: p1.x + (p2.x - p1.x) / 2,
            y: p1.y + (p2.y - p1.y) / 2
        },
        pC = {
            x: p0.x + (p2.x - p0.x) / 2,
            y: p0.y
        };

        /*Dibujar los tres triángulos intermedios*/
        this.drawFract(p0, pA, pC, nivel - 1);
        setTimeout(() => this.drawFract(pA, p1, pB, nivel - 1), this.delay1);
        setTimeout(() => this.drawFract(pC, pB, p2, nivel - 1), this.delay2);
    } else {
        /*Triángulo principal*/
        this.drawTriangle(p0, p1, p2);
    }
}

/*Activar animación*/
animacion(check) {
    if (check === true) {
        this.delay1 = 500;
        this.delay2 = 1000;
    } else {
        this.delay1 = 0;
        this.delay2 = 0;
    }
}

/*Limpia la imagen y vuelve a dibujar con el nivel especificado*/
actualizarFractal = (data) => {
    let val = data.target.value;
    document.getElementById("textInput").value = val;
    document.getElementById("niveles").value = val;
    document.getElementById("cant").innerHTML = 3 ** parseInt(val);

    this.ctx.clearRect(0, 0, 401, 401);
    this.drawFract({ x: 0, y: 400 }, { x: 200, y: 0 }, { x: 400, y: 400 }, val);
}

```

Прогнозування на основі теорії фракталів.

Опис програми

УКР.НТУУ"КПІ ім. Ігоря Сікорського"\_ТЕФ\_АПЕПС\_ТМ52\_19Б 13-2

Аркушів 7

Київ 2019

## **АНОТАЦІЯ**

Інтерфейс створеного програмного забезпечення по прогнозування на основі теорії фракталів дозволяє користуватися системою з будь-якого електронного пристрою. Програма забезпечує якісне прогнозування графіку на основі

фрактального аналізу та дозволяє будувати різні фрактали.

Додаток було розроблено за допомогою платформи JetBrains PhpStorm 2018. З використанням мови JavaScript та фреймворку Angular на клієнтській стороні, технологія Deploid на серверній стороні. В якості бази даних було використано MongoDB.

# **ЗМІСТ**

1. ЗАГАЛЬНІ ВІДОМОСТІ
2. ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ
3. ОПИС ЛОГІЧНОЇ СТРУКТУРИ
4. ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ
5. ВХІДНІ ТА ВИХІДНІ ДАНІ

## ЗАГАЛЬНІ ВІДОМОСТІ

Відповідно до теми дипломної роботи, програма має назву «Прогнозування на основі теорії фракталів».

Програма працює через браузер з будь-якого електронного пристрою та потребує доступу до мережі інтернет. Вона забезпечує практичність та швидкість в роботі.

Програмне забезпечення було написане мовою JavaScript з використанням фреймворку Angular на клієнтській стороні, технологія Deploid на серверній стороні. В якості бази даних було використано MongoDB.



## **ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ**

Розроблений програмний засіб покликаний покращити прогноз економічних процесів, а саме - забезпечити зручне та ефективне прогнозування фінансових ринків на основі теорії фракталів. Це було реалізовано за допомогою наступного функціоналу:

- створення графіку;
- створення / редагування / видалення / перегляд компаній;
- прогнозування графіку за допомогою фрактального аналізу;
- прогнозування тренду;
- побудова фракталу Серпинського з різною кількістю трикутників;
- створення різних видів фракталів;
- перегляд загальної інформації про компанії;
- можливість перегляду аналізу в зручному вигляді з будь-якого електронного пристрою з доступом до мережі інтернет;.

## ОПИС ЛОГІЧНОЇ СТРУКТУРИ

При відкритті сторінок дані для відображення завантажуються одноразово: посилається запит на сервер, сервер посилає запит на базу даних повертаючи результат на клієнтську сторону, де в залежності від результату оновлюється користувацький інтерфейс. Дані завантажуються тільки тоді, коли відбулися певні зміни в базі даних.

## **ТЕХНІЧНІ ЗАСОБИ ЩО ВИКОРИСТОВУЮТЬСЯ**

Для організації доступу до програмного продукту потрібно мати будь-який електронний пристрій.

Кінцевим користувачам для роботи з програмою потрібен браузер та доступ до мережі інтернет.

## **ВХІДНІ І ВИХІДНІ ДАНІ**

Вхідними даними є:

- Статичні данні з графіків різних компаній.
- коефіцієнти Херста для аналізу графіка;

Вихідними даними є:

- змодельований графік;
- змодельований тренд;
- змодельований прогноз ціни в подальший час;